****

**Universidade Anhanguera Uniderp**

**Engenharia da Computação**

**USO DE BASES METEREOLÓGICAS RELACIONAIS EM BANCO DE DADOS NOSQL PARA GERAR INFORMAÇÕES QUE APOIEM A TOMADA DE DECISÃO DE AGRICULTORES**

**Campo Grande/MS**

**MAI/2015**

**João Rafael Campos da Silva – RA 1099477936**

**USO DE BASES METEREOLÓGICAS RELACIONAIS EM BANCO DE DADOS NOSQL PARA GERAR INFORMAÇÕES QUE APOIEM A TOMADA DE DECISÃO DE AGRICULTORES**

Monografia apresentada como exigência parcial para obtenção do grau de Bacharelado em Engenharia da Computação da Universidade Anhanguera Uniderp.

Orientador: Prof.ª MSc. Edilene Veneruchi de Campos

**Campo Grande/MS**

**MAI/2015**

Dedico este trabalho a minha família, a minha namorada, aos meus amigos, meu orientador e a todos os professores que participaram do meu desenvolvimento acadêmico; pois foram de suma importância ao longo dessa graduação.

**AGRADECIMENTOS**

A esta universidade, seu corpo docente, direção e administração que não pouparam esforços para me ajudar na hora das dificuldades.

Ao meu orientador, pelo suporte no pouco tempo que lhe coube, pelas suas correções e incentivos.

Aos meus professores que forneceram todo o conhecimento necessário para que eu conseguisse realizar a construção desse protótipo.

Aos meus pais por nunca deixarem de me apoiar durante toda a minha graduação.

E a todos que direta ou indiretamente me ajudaram nessa conquista tão importante para a minha vida.

**RESUMO**

Com a grande popularização da internet, o mundo está ficando cada vez mais digital e com isso a quantidade de dados existentes sobre diversos ramos cresce a cada dia. Um dos setores que se informatizou há um bom tempo e acumula quantias gigantescas de dados hoje em dia é a meteorologia. Para o processamento desse grande volume de dados de uma forma rápida foram surgindo várias tecnologias que antes não eram tão importantes, como banco de dados NoSQL e modelos de processamento distribuído como o Hadoop. O projeto abordado nos próximos capítulos visa fundamentar essas tecnologias para que seja possível armazenar uma grande massa de dados meteorológicos obtidas de bases relacionais como PROTIN e BDMEP em um banco de dados NoSQL a fim de gerar informações precisas com maior rapidez que possam auxiliar agricultores na tomada de decisão.

**Palavras-chave:** Big Data, Hadoop, NoSQL, WebService, Agricultura, Banco de Dados, Meteorologia.

**ABSTRACT**

With the wide popularization of the internet, the world is becoming increasingly digital and with it the amount of existing data on various branches grows every day, one of the sectors that if Quicken for a long time and accumulate huge amounts of data these days is the weather. For the processing of the large volume of data quickly arose several technologies that before were not as important as NoSql database and distributed processing models like Hadoop. The project addressed in later chapters seeks to substantiate these technologies so that you can store a large mass of weather data obtained from relational databases as DINESH and BDMEP in a NoSql database in order to generate accurate information faster that can help farmers in decision-making.

**Keywords:** BigData, Hadoop, NoSql, WebService, Agriculture, Database, Meteorology.

**LISTA DE FIGURAS**

[Figura 1 - Gráfico demonstrativo Big Data](#_Toc793086447) 13

[Figura 2 - Linha de Processamento do BigData](#_Toc260721147) 14

[Figura 3 - Infraestrutura Apache Hadoop](#_Toc1396107303) 17

[Figura 4 - Página de Cadastro do BDMEP](#_Toc1396107303) 29

[Figura 5 - Página de Login do BDMEP](#_Toc1396107303) 30

[Figura 6 - Menu de Consulta do BDMEP](#_Toc1396107303) 30

[Figura 7 - Página de Consulta Avançada do BDMEP](#_Toc1396107303) 31

[Figura 8 - Página de Resultados do BDMEP](#_Toc1396107303) 31

[Figura 9 – Dados da Consulta realizada no BDMEP](#_Toc1396107303) 32

[Figura 10 - Página de Consulta do PROTIM](#_Toc1396107303) 32

[Figura 11 – Dados da Consulta realizada no PROTIM](#_Toc1396107303) 33

[Figura 12 – Implementação da permissão de acesso ao Hadoop](#_Toc1396107303) 34

[Figura 13 - Aba de Transferência de Arquivos para o HDFS](#_Toc1396107303) 35

[Figura 14 - Abas para listar, remover e ler Arquivos ou Diretórios no HDFS](#_Toc1396107303) 35

[Figura 15 – Função de Mapeamento](#_Toc1396107303) 36

[Figura 16 – Função de Redução](#_Toc1396107303) 37

[Figura 17 – Permissões para executar ações no HDFS](#_Toc1396107303) 38

[Figura 18 – Propriedades para executar o MapReduce no HDFS](#_Toc1396107303) 39

[Figura 19 – Resultado gerado pelo MapReduce](#_Toc1396107303) 39

**LISTA DE ABREVIATURAS E SIGLAS**

|  |  |
| --- | --- |
| BDMEP  HDFS  HTTP  INMET  NOSQL  PROTIM  REST | Bando de dados Meteorológicos para Ensino e Pesquisa  Hadoop Distributed File System  Hypertext Transfer Protocol  Instituto Nacional de Meteorologia  Not Only SQL  Portal de Tecnologia da Informação para Meteorologia  Representational State Transfer |
|  |  |
|  |  |
|  |  |
|  |  |

**SUMÁRIO**

[**1 INTRODUÇÃO 9**](#_Toc435707036)

[**2 OBJETIVOS 10**](#_Toc435707037)

[2.1 Objetivo Geral 10](#_Toc435707038)

[2.2 Objetivos Específicos 10](#_Toc435707039)

[**3 ESTRUTURA DE CAPÍTULOS 11**](#_Toc435707040)

[**4 BIG DATA 12**](#_Toc435707041)

[4.1 Conceitos de Big Data 12](#_Toc435707042)

[4.2 Etapas do processo Big Data 13](#_Toc435707043)

[4.3 Vantagens 14](#_Toc435707044)

[**5 APACHE HADOOP 16**](#_Toc435707045)

[5.1 Fundamentos do Apache Hadoop 16](#_Toc435707046)

[5.2 Introdução ao MapReduce 17](#_Toc435707047)

[5.3 Sistema de Arquivo HDFS 18](#_Toc435707048)

[5.3.1 NameNode 18](#_Toc435707049)

[5.3.2 DataNode 19](#_Toc435707050)

[**6 BANCOS DE DADOS NOSQL 20**](#_Toc435707051)

[6.1 Introdução ao NoSQL 20](#_Toc435707052)

[6.2 Comparando Bancos NoSQL com o Modelo Relacional 21](#_Toc435707053)

[6.3 Características de Bancos NoSQL 22](#_Toc435707054)

[6.3.1 Bancos de Dados Chave/Valor 22](#_Toc435707055)

[6.3.2 Bancos de Dados Orientados a Documentos 23](#_Toc435707056)

[6.3.3 Bancos de Dados de Famílias de Colunas 23](#_Toc435707057)

[6.3.4 Bancos de dados de grafos 24](#_Toc435707058)

[**7 WEBSERVICE REST 25**](#_Toc435707059)

[7.1 Conceitos de REST 25](#_Toc435707060)

[7.1.1 Semânticas de Recursos 25](#_Toc435707061)

[7.1.2 Interação por Métodos 26](#_Toc435707062)

[**8 METODOLOGIA 27**](#_Toc435707063)

[**9 DESENVOLVIMENTO E RESULTADOS 29**](#_Toc435707064)

[9.1 Importando de Dados Meteorológicos do BDMEP 29](#_Toc435707065)

[9.2 Importando Dados Meteorológicos do PROTIM 32](#_Toc435707066)

[9.3 Manipulando Arquivos no HDFS 33](#_Toc435707067)

[9.4 Testando o MapReduce com dados de Temperatura 36](#_Toc435707068)

[**10 CONCLUSÃO 40**](#_Toc435707069)

[**REFERÊNCIAS BIBLIOGRÁFICAS 41**](#_Toc435707070)

# 1 INTRODUÇÃO

Busca-se fundamentar nesse trabalho tecnologias relacionadas a construção de uma infraestrutura para o Big Data. Serão abordados conceitos de Big Data assim como a introdução a sua infraestrutura. Discutiremos também, opções de bancos de dados NoSQL, que são compatíveis com o modelo distribuído do Big Data, e webservices REST para a coleta dos dados de bancos de dados meteorológicos relacionais.

Bases de dados meteorológicos como PROTIN e BDMEP armazenam diariamente em forma digital uma infinidade de informações, referentes as séries históricas da rede de estação do INMET, são 291 estações meteorológicas convencionais que armazenam cerca de 3 milhões de informações sobre o clima (BDMEP, 2015).

Atualmente esses dados se encontram em bancos de dados relacionais, onde a busca por alguma informação de valor pode ser demorada devido a grande massa de dados. Uma alternativa seria realizar uma modelagem especial com bancos de dados não relacionais, permitindo uma escalabilidade mais barata e menos trabalhosa além da rapidez em que esses dados poderiam ser acessados por um sistema analítico fornecendo informações valiosas que em bancos relacionais talvez não seriam possíveis.

Isso tudo se encaixa no termo BigData que de forma geral consiste em um grande volume de dados não estruturados que se processados com agilidade e com um objetivo em mente podem servir de grande apoio a tomadas de decisões (DEVMEDIA, 2015). Dados meteorológicos armazenados diariamente desde 1961, se processados com essa rapidez, poderiam gerar uma previsão das melhores épocas para que um agricultor realize um plantio seguro (BDMEP, 2015).

# 2 OBJETIVOS

## 2.1 Objetivo Geral

Auxiliar agricultores na tomada de decisão baseada em uma base de dados meteorológicos não estruturada, gerada a partir de um banco de dados relacional como BDMEP e PROTIN.

## 2.2 Objetivos Específicos

* Encontrar na internet grandes bases de dados meteorológicas, tais como BDMEP e PROTIN.
* Entender o processo de geração de um banco de dados NoSql a partir de bancos de dados estruturados como BDMEP e PROTIN.
* Pesquisar e estudar web services e algoritmos para a comunicação com bancos de dados meteorológicos estruturados fornecidos pelo governo.
* Estudar, entender e realizar uma modelagem especial NoSql para o armazenamento do grande volume de dados obtido.
* Identificar problemas reais que agricultores possuem para a tomada de decisão baseada na meteorologia.
* Identificar softwares, algoritmos ou estratégias de tratamento do Big Data que podem ser utilizadas no projeto.
* Realizar a construção de um website protótipo para a consulta do clima em um determinado espaço de tempo analisando os dados a partir de um banco de dados NoSql.

# 3 ESTRUTURA DE CAPÍTULOS

# 

No primeiro momento são abordados conceitos do termo Big Data, exemplificando quais são as etapas para o desenvolvimento de uma aplicação para esse meio, assim como as vantagens de se utilizar dessa tecnologia.

Depois falaremos de uma ferramenta da empresa Apache usada para tratar o Big Data chamada Hadoop. Serão apresentados fundamentos e como essa estrutura é montada, assim como o algoritmo MapReduce implementado por ela e seu sistema de arquivos chamado HDFS.

Para o armazenamento de dados considerados grandes o bastante para se tornarem um Big Data é recomendado que sejam utilizados bancos de dados não estruturados, conhecidos hoje em dia como NoSql. Serão apresentadas a características desse tido de banco e quais são as peculiaridades de cada tipo.

Para estabelecer uma comunicação segura entre a aplicação apresentada ao usuário e a estrutura do Hadoop para o processamento dos dados, serão utilizados webservices Rest. O assunto é abordado fundamentando suas características e exemplificando o seu uso.

Para encerrar serão expostas as metodologias empregadas na construção do projeto.

# 4 BIG DATA

Com o grande desenvolvimento da internet no mundo, muitas coisas estão sendo digitalizadas. Nesse capitulo serão levantados conceitos de um novo sistema de tratamentos para esse grande volume de dados gerados diariamente.

## 4.1 Conceitos de Big Data

Em meio ao grande crescimento da tecnologia no mundo as grandes organizações preveem que o grande volume de dados gerados pela internet servirá de grande ajuda no apoio a decisões baseada em estatísticas. Os dados acumulados são oriundos de redes sociais, sensores, e-mails, vídeos, websites, sensores, câmeras e smartphones, entre outros. (PARAIZO, 2013)

“O Facebook divulgou ao blog TechCrunch que em 2012, que processa 2,5 bilhões de conteúdo e mais de quinhentos terabytes de dados por dia. São muitos dados se comparados há alguns anos. Em 2000 25% dos dados estavam em formato digital e em 2007 já eram 94%, ou seja, um crescimento de 69% em sete anos.” (DEVMEDIA, 2014, p.1)

Apesar de todo esse conteúdo estar disponível na internet para qualquer empresa utilizar, a maioria deles não é tratado e analisado de uma forma que forneça informações uteis para a organização. Por esse motivo o Big Data tem chamado a atenção de vários analistas de dados, o mesmo não possui uma definição exata, depende da situação em que é aplicado.

“Big Data são tecnologias e práticas emergentes que possibilitam a seleção, processamento, armazenamento e geração de insights de grandes volumes de dados estruturados e não estruturados de maneira rápida, efetiva e a um custo acessível. Big Data pode ser considerado como um conjunto de dados que cresce exponencialmente e necessita de habilidades além das quais as ferramentas típicas de gerenciamento e processamento de informações dispõem. Taurion descreve ainda que se trata de "Um conjunto de tecnologias, processos e práticas que permitem às empresas analisarem dados a que antes não tinham acesso e tomar decisões ou mesmo gerenciar atividades de forma muito mais eficiente”. Segundo o autor citado acima, não é teoria ou futurologia, é algo que se encontra agora.” (DEVMEDIA, 2014, p.1)

O Big Data pode ser definido mais simplificadamente por “Big Data = volume + variedade + velocidade + veracidade, gerando algum valor”. Quando falamos em volume, estamos nos referindo a grande massa de dados disponíveis na internet que pode ser utilizada de modo oportuno por qualquer organização que se interesse. Por se tratar de dados originários de diversos lugares, a variedade de informações obtidas é imensa. Para que a empresa possa obter algum lucro do investimento aplicado no processamento desses dados, o retorno de informações uteis deve ser ágil e os mesmos devem ser confiáveis, caso contrário, as informações se tornam lixo acumulado em servidores caros, prejudicando a organização. (TAURION, 2013)

|  |
| --- |
| C:\Users\jrcsilva\Desktop\1.1.jpg |
| **Figura 1 - Gráfico demonstrativo Big Data** |
| **Fonte: (KNXISAAC, 2014).** |

Como a ampliação do termo Big Data, várias empresas vêm tomando iniciativas para se beneficiar desse novo conceito. Mas não é tão simples assim, a organização deve elaborar uma estratégia bem definida e estar cientes que acoplagem do Big Data pode implicar em serias mudanças na área de negócio e infraestrutura de tecnologia.

## 4.2 Etapas do processo Big Data

Para que a implantação do conceito Big Data seja implantada com sucesso em uma empresa, devem ser seguidas algumas etapas de processamento.

A primeira fase do Big Data é a coleta de dados, deve-se coletar o maior volume e variedade de informações e armazená-las para futuramente serem processadas. Nessa fase também é necessário que dados quebrados ou sem sentido sejam descartados, para que não influenciem em uma análise posterior. (CHEDE, 2012)

Depois disso vem a parte de integração, agregação e representação dos dados obtidos. Cada tipo de dado deve receber um tratamento diferente, esse objetivo pode ser alcançado com mais facilidade se forem elaboradas categorias para cada um dos tipos de dados encontrados. (CHEDE, 2012)

A próxima etapa é realizar modelagem desses dados, por se tratar de uma variedade enorme de categorias requer um conhecimento maior sobre as regras de negócio, aqui entra o “Cientista de Dados”.

"Cientista de Dados é um profissional com habilidades em ciência da computação, matemática, estatística e conhecimento de negócio." (ALGARTECH, 2012, p.1)

A última fase propõe a realização de uma pesquisa sobre novas formas de visualização, que irão aliciar na interpretação dos dados processados e a obtenção de informações úteis que ajudem de alguma forma no desenvolvimento da empresa.

|  |
| --- |
|  |
| **Figura 2 - Linha de Processamento do BigData** |
| **Fonte: (DEVMEDIA, 2014).** |

## 4.3 Vantagens

A seguir é apresentada uma lista com as principais vantagens da utilização do Big Data:

• Compreender o que os clientes realmente necessitam, estudando seus hábitos de consumo. Quando se entende exatamente o que p cliente quer, o resultado obtido pode ser mais claro, ganhando a confiança do mesmo. (HABITZREITER, 2014)

• Encontrar potenciais compradores a partir da mensuração em tempo real das redes sociais. O desenvolvimento da tecnologia permite que pessoas de diversas localidades geográficas conheçam o produto e ofertas em tempo real, com isso pode ocorrer expansão nas vendas (HABITZREITER, 2014)

• Evitar que haja imprevistos no mercado graças a análise em tempo real de informações de diversas variáveis do mercado. Tudo pode ser analisado em tempo real, tornando possível a prevenção e antecipação a possíveis dificuldades no mercado e tomar decisões para contornar esses problemas. (ZUPPO, 2014)

• É possível monitorar a concorrência sem que ela perceba para criar ofertas especiais. Conhecer o concorrente e extremamente necessário para aumentar lucros. (ZUPPO, 2014)

Estas são as principais vantagens de se utilizar Big Data, existem outras, pois como disse anteriormente, o Big Data se trata de vários conceitos.

# 5 APACHE HADOOP

Neste capítulo é abordado um framework da Apache que vem sendo a principal escolha de grandes empresas na hora de montar uma infraestrutura para trabalhar com Big Data.

## 5.1 Fundamentos do Apache Hadoop

Estamos vivendo a era da informação onde a geração de dados digitais é cada vez maior, alcançando os petabytes diários caracterizando o Big Data.

No entanto, é necessária uma implementação que analise esse grande volume de dados e processe pesados cálculos para identificar comportamentos e fornecer sugestões.

Para solucionar esse problema surgiu o Apache Hadoop capaz de processar muitos dados em um sistema computacional distribuído. O framework também contém um modelo de programação e um sistema de armazenamento de dados que permitem um processamento muito rápido.

O Haddop possui um modelo de programação chamado de MapReduce que permite o processamento distribuído de uma grande tarefa. Além do MapReduce o framework também contém um sistema de armazenamento chamado de HDFS (Hadoop Distributed File System) que como o próprio nome diz, também trabalha de forma distribuída facilitando o trabalho do MapReduce. (GASPAROTTO, 2013)

Ao longo do tempo foram incorporadas outras ferramentas ao Hadoop como o Hive, Pig, ZooKeeper, Flume, Mahout entre outras que buscam solucionar problemas específicos na arquitetura agregando ainda mais o framework Hadoop. (GASPAROTTO, 2013)

O Haddop foi criando para trabalhar de forma distribuída, mas também é possível usá-lo de forma independente ou pseudo distribuído facilitando para o desenvolvedor na hora da implementação das funcionalidades. (MARTINS, 2013)

|  |
| --- |
|  |
| **Figura 3 - Infraestrutura Apache Hadoop** |
| **Fonte: (MARTINS, 2013).** |

## 5.2 Introdução ao MapReduce

Hoje em dia é um desafio armazenar, manipular e analisar uma grande quantidade de dados de forma inteligente. Sistemas corporativos ou sociais geram milhares de dados por dia que não podem ser processados por tecnologias tradicionais.

Pensando nesse contexto surgiu o termo Big Data, que é definido pelos "3v", volume, velocidade e variedade. (ALECRIM, 2013)

Diante desse desafio, dois engenheiros do Google desenvolveram uma ferramenta chamada MapReduce que permite separar uma grande massa de dados em pequenos pedaços e distribuir o processamento dos mesmos em vários computadores de baixo custo, tornando o buscador da empresa mais rápido e eficiente. (ALECRIM, 2013)

O MapReduce se mostrou adequado para trabalhar com problemas que podem ser divididos em subproblemas. Ele é um modelo de programação que une o Map, mapeamento de dados, que serve como entrada para a função Reduce que minimiza esses dados em informações úteis para a empresa. (ANDRADE, 2013)

Vários Maps e Reduces podem ser espalhados em maquinas de baixo custo tornando o processamento ágil e escalável. Para a utilização do MapReduce é necessário que a entrada de dados esteja organizada em uma lista contendo chave e valor para que possam ser dividas para o processamento paralelizado. (ANDRADE, 2013)

## 5.3 Sistema de Arquivo HDFS

O Hadoop possui seu próprio sistema de arquivos distribuído, esse sistema é chamado de HDFS (Hadoop Distributed File System), ele é responsável pela organização, armazenamento, localização, compartilhamento e proteção de arquivos que ficam distribuídos em uma rede de servidores. (HANSON, 2013)

Em um sistema distribuído é essencial que haja um controle de concorrência, ou seja, quando diversos usuários tentam gravar o mesmo arquivo. Tal controle é necessário para que haja uma operação atômica e não prejudique a consistência nas informações obtidas. Um sistema que trabalha de forma distribuída deve garantir a atomicidade nas operações de leitura, escrita, criação ou remoção de um arquivo, de forma transparente para quem manipula os dados. (HANSON, 2013)

Segundo HANSON o HDFS faz parte da camada de armazenamento do Hadoop, o mesmo foi otimizado para operações de leitura e escrita visando o processamento de arquivos grandes na escala de Terabytes que ficam localizados em computadores de um cluster.

O sistema de arquivos do Haddop é altamente escalável e disponível, graças a replicação de dados em vários computadores e um processo de tolerâncias a falhas. O HDFS se encarrega dividir os arquivos em blocos, normalmente de 64MB, e espalha-los por várias maquinas (Em caso de cluster no mínimo 3 e em modo local no minimo 1) ficando assim tolerante a falhas tanto de software quanto de hardware. O HDFS é dividido estruturado em um padrão mestre-escravo com dois processos principais, NameNode e DataNode. (MICROSOFT, 2014)

### 5.3.1 NameNode

O NameNode tem a função de gerenciar arquivos armazenados no HDFS, guardando informações sobre quais DataNodes são responsáveis por quais blocos de dados de cada arquivo, organizando as informações em uma tabela de metadados. (JONES, 2010)

“São funções do NameNode mapear a localização, realizar a divisão dos arquivos em blocos, encaminhar os blocos aos nós escravos, obter os metadados dos arquivos e controlar a localização de suas réplicas.” (MARTINS, 2013, p.1)

O NameNode é constantemente acessado e é necessário um desempenho aceitável, por esse motivo, todas as informações são mantidas em memória. Ele integra o HDFS e está localizado no nó mestre da aplicação. (JONES, 2013)

### 5.3.2 DataNode

O DataNode, como parte do nome diz, fica responsável pelo armazenamento do conteúdo dos arquivos nos computadores escravos. Como o HDFS é um sistema de arquivos distribuído, é normal que existirem diversas instâncias de DataNode em uma aplicação Hadoop, possibilitando que os arquivos sejam particionados em blocos que ficam replicados em computadores diferentes. Um DataNode poderá guardar diversos blocos, inclusive de diferentes arquivos, no entanto, eles precisam retornar informações constantemente ao NameNode, comunicando-o sobre as operações que estão sendo executadas nos blocos. (SILVA, 2015)

# 6 BANCOS DE DADOS NOSQL

Nesse capitulo serão abordados conceitos de NoSQL e porque ele é associado ao Big Data introduzindo quais são os principais Bancos dessa categoria usados atualmente.

## 6.1 Introdução ao NoSQL

O termo NoSQL surgiu em 1998 como alternativa de um banco de dados não relacional de código aberto. Segundo Carlo Strozzi, seu autor, alega que a iniciativa NoSQL "é completamente distinto do modelo relacional e portanto deveria ser mais apropriadamente chamado "NoREL" ou algo que produzisse o mesmo efeito". (IANNI, 2012)

Com o grande avanço da internet e o surgimento de novas tecnologias foram aparecendo novos tipos de dados mais complexos que se tornaram uma dificuldade para o modelo relacional e com uma manutenção muito cara. Em 2006 o termo NoSql voltou a ser comentado. (PORCELLI, 2010)

“Em 2006, o artigo: BigTable: A Distributed Storage System for Structured Data, publicado pelo Google em 2006, traz novamente à tona o conceito NoSQL. No início de 2009, o termo NoSQL é reintroduzido por um funcionário do Rackspace, Eric Evans, quando Johan Oskarson da Last.fm queria organizar um evento para discutir bancos de dados open source distribuídos.” (IANNI, 2012, p.1)

O nome NoSQL era uma tentativa de caracterizar diversos bancos de dados não relacionais que estavam surgindo tentando fazer referência aos nomes de bancos de dados relacionais populares como MySQL, MS SQL, PostgreSQL, etc. (ITEXTO, 2012)

Com a popularização das redes sociais e telefonia móvel, a internet se tornou uma fonte imensa de dados não estruturados que se encaixaram perfeitamente no modelo não relacional, ou seja, bancos NoSQL. (ITEXTO, 2012)

Portanto, os bancos de dados NoSQL, foram ficando mais populares entre as poderosas empresas pois reúnem as particularidades de poder trabalhar com dados semiestruturados ou não estruturados vindos de variadas origens (arquivos multimídia, arquivos de log, websites, etc.). (WEISSMANN, 2015)

## 6.2 Comparando Bancos NoSQL com o Modelo Relacional

Por muito tempo vem sendo usado bancos de dados relacionais como a principal escolha para o armazenamento de dados corporativos, porem grandes empresas vem adotando uma nova forma de tratar esses dados.

Com o grande aumento no volume de dados coletados diariamente e a previsão de que isso só tem a crescer, bancos relacionais perdem em escalabilidade.

Como alternativa para solucionar esse problema surgiu o NoSql, que vem do termo Not Only SQL que proporciona a possibilidade de trabalhar em um ambiente distribuído, aumentando a escalabilidade e oferecendo um alto desempenho a baixo custo. (WEISSMANN, 2012)

Para solucionar o problema de escalabilidade o modelo relacional geralmente são utilizadas duas alternativas, o aumento do número de servidores (escalabilidade horizontal) ou o upgrade dos servidores (escalabilidade vertical). Mas se o volume de dados continuar a crescer rapidamente essa alternativa se torna muito custosa. Uma melhor solução seria trabalhar de forma distribuída, no entanto, o sistema estruturado em linhas e colunas dos modelos relacionais gera uma enorme dificuldade no manuseio desses dados quando as tabelas se encontram em computadores diferentes. Por conta disso foram surgindo as soluções não relacionais. (SOUZA, 2012)

O NoSQL busca uma forma nova de modelagem, modificando a rígida e tradicional estrutura utilizada nos bancos relacionais. Grandes empresas tinham como objetivo flexibilizar os dados conforme as suas necessidades, essa flexibilidade é extremamente importante para o acesso distribuído, fornecendo alta escalabilidade e dispensabilidade para as aplicações. (SOUZA, 2012)

A intenção do NoSQL não é substituir o modelo relacional tradicional, mas sim fornecer uma alternativa para as empresas que tratam de um grande volume de dados e que os mesmos possam ser um pouco mais flexíveis. (BRITO, 2012)

Os bancos de dados NoSQL são divididos da maneira que armazenam as informações, os principais tipos são: chave-valor, colunas, grafos e documentos. (BRITO, 2012)

Por tanto o NoSQL vem para somar como uma alternativa altamente escalável e com grande disponibilidade de dados consistentes, portanto vale analisar bem e escolher a melhor alternativa de Banco de dados para a sua empresa.

## 6.3 Características de Bancos NoSQL

Segundo o portal de artigos DEVMEDIA, essas são as principais características dos bancos de dados NoSQL.

Bancos de dados NoSQL visão a utilização de processamento paralelo para a manipulação de informações. Para que o processamento de um grande volume de dados ocorra com uma performance razoável, é mais eficaz que o trabalho seja dividido em várias tarefas menores e que podem assim ser executadas de forma assíncrona utilizando todo o poder de processamento de cada máquina em uma infraestrutura distribuída.

O uso de vários processadores pequenos trabalhando ao mesmo tempo se torna mais eficiente do que um processador maior que pode ficar ocioso executando uma única função de cada vez, além de ser uma alternativa muito mais econômica evitando que empresas fiquem presas as poucas fornecedoras que disponibilizam hardwares poderosos.

Para atender seus usuários de forma mais apropriada as empresas distribuem seus data centers por diversas partes do país ou do mundo, visando solucionar problemas de disponibilidade e performance. (KOKAY, 2012)

A distribuição deles combinada com o hardware barato, obrigando ao sistema a necessidade de ser sólido o suficiente para tolerar falhas constantes e imprevisíveis, seja a mesma relacionada ao hardware ou a infraestrutura do local onde o data center se localiza. (KOKAY, 2012)

Pensando nessas necessidades foram surgindo diversos tipos de bancos NoSQL buscando atender a diversas particularidades.

### 6.3.1 Bancos de Dados Chave/Valor

Sistemas organizados nessa categoria, conhecidos também como tabelas de hash distribuídas, armazenam objetos indexados por chaves, e possibilitam a busca por esses objetos a partir de suas chaves. (VARDANYAN, 2012)

Alguns bancos que utilizam esse padrão são: DynamoDb, Couchbase, Riak, Azure Table Storage, Redis, Tokyo Cabinet, Berkeley DB, etc. (VARDANYAN, 2012)

### 6.3.2 Bancos de Dados Orientados a Documentos

Os documentos dos bancos dessa categoria são coleções de atributos e valores, onde o atributo pode ser multivalorado. Normalmente os bancos de dados orientados a documento não possuem tipo definido, ou seja, os documentos armazenados não têm a mesma estrutura. (MOURA, 2012)

Esse aspecto os torna ótimas opções para o armazenamento de dados semi estruturados. Alguns bancos que utilizam esse padrão são: MongoDb, CouchDB, RavenDb, etc. (MOURA, 2012)

### 6.3.3 Bancos de Dados de Famílias de Colunas

Bancos de dados relacionais frequentemente armazenam os registros das tabelas contiguamente no disco. Por exemplo, caso se queira guardar id, usuário e senha de pessoas em um sistema comercial, os registros seriam: (PERERA, 2012)

Id1, Usuario1, Senha1.

Id2, Usuario2, Senha2.

Esse modo de estruturar os dados possibilita uma escrita muito rápida, pois todos os registros são armazenados no disco em uma única operação de escrita no banco. Essa estrutura é igualmente eficaz caso se precise realizar a leitura de registros inteiros.

Mas para situações onde se quer ler algumas poucas colunas de muitos registros, essa estrutura é pouco eficiente, pois muitos blocos do disco terão de ser lidos. No entanto, quando é necessário fazer a leitura de poucas colunas em uma tabela com muitos registros, essa estrutura se torna ineficiente. Quando é necessário otimizar a leitura de dados estruturados, os bancos de dados de famílias são uma escolha mais competente, pois os mesmos guardam os dados contiguamente por colunas. (PERERA, 2012)

O Se utilizarmos o exemplo anterior em um banco de dados dessa categoria ficaria: Id1, Id2; Usuario1, Usuario2; Senha1, Senha2.

Uma desvantagem dessa estrutura é a inserção de um novo registro que se torna bem mais custosa do que em um banco de dados relacional tradicional. Assim os bancos tradicionais são mais eficientes em sistemas para processamento de transações online (OLTP), enquanto os bancos em famílias de colunas são mais adequados para processamento analítico online (OLAP). Alguns bancos de dados que são orientados a coluna: Hypertable, Cassanda, Amazon SimpleDB, Hadoop, etc. (PERERA, 2012)

### 6.3.4 Bancos de dados de grafos

Ao contrario dos outros tipos de bancos de dados NoSQL, esse está rigorosamente vinculo ao modelo de grafos. A intensão da estrutura é representar os dados no esquema como grafos dirigidos, ou estruturas que se pareçam com elas. (SATO, 2012)

"O modelo de grafos é mais interessante que outros quando informações sobre a interconectividade ou a topologia dos dados são mais importantes, ou tão importante quantos os dados propriamente ditos." (IANNI, 2012, p.1)

O banco orientado a grafos possui três componentes: os nós (são os vértices do grafo), os relacionamentos (são as arestas) e as propriedades (ou atributos) dos nós e relacionamentos. No caso, o banco de dados pode ser observado como um multigrafo rotulado e direcionado, onde cada par de nós pode ser conectado por mais de uma aresta. Alguns bancos que utilizam esse padrão são: Neo4J, Infinite Graph, InforGrid, HyperGraphDB, etc. (SATO, 2012)

# 7 WEBSERVICE REST

## 7.1 Conceitos de REST

O REST é fortemente baseado no protocolo HTTP que é basicamente a base da Web como a conhecemos, sendo que a navegação desse protocolo pode ser considerado REST. Porém o REST não é tão simples quanto o protocolo HTTP, ele é constituído por várias regras que devem ser seguidas para que se possa utilizar o protocolo. (SAUDATE, 2013)

### 7.1.1 Semânticas de Recursos

Basicamente todos os serviços e operações executados pelo REST são baseados em recursos, que são entidades bem definidas em softwares, que possuem endereços próprios e identificadores. (XAVIER, 2013)

Como exemplo podemos referenciar um sistema de comercio de camisetas e a camiseta é um recurso. Assim, segundo as regras do REST dizem que camisetas devem ter uma URL própria e que esta URL deve ser significativa. Desta forma uma URL que deixaria claro o recurso camisetas seria /camisetas. (XAVIER, 2013)

De acordo como modelo REST, esta URL realizará interação com todas as camisetas do sistema. Para tratar de camisetas específicas, são usados identificadores. Estes identificadores podem ter qualquer formato, por exemplo, suponha uma camiseta Abercombie com uma chave primária 123456. Note que o nome da camiseta não serve como identificador, por existirem várias dessa marca, mas a chave primaria sim, por ser única em todo o banco de dados. Desta forma, deve ser possível buscar esta camiseta através da sua chave primária, com a URL /camisetas/123456. Lembre-se, regras semelhantes aplicam-se a qualquer recurso. O identificador a ser utilizado na URL pode ser qualquer coisa que você assim desejar, e mais de um tipo de identificador pode ser usado para alcançar um recurso. (SAUDATE, 2013)

### 7.1.2 Interação por Métodos

Para interagir com as URL’s, os métodos HTTP são utilizados. A regra de ouro para esta interação é que URL’s são substantivos, e métodos HTTP são verbos. Isto quer dizer que os métodos HTTP são os responsáveis por provocar alterações nos recursos identificados pelas URL’s. (TILKOV, 2008)

Estas modificações são padronizadas, de maneira que:

• GET - recupera os dados identificados pela URL

• POST - cria um novo recurso

• PUT - atualiza um recurso

• DELETE - apaga um recurso

Note que estes quatro métodos principais podem ser diretamente relacionados a operações de bancos de dados. Assim, para recuperar o cliente de número 1 do banco de dados, basta utilizar o método GET em conjunto com a URL /camiseta/1; para criar um novo cliente, basta utilizar o método POST sobre a URL /camiseta (o identificador será criado pelo banco de dados); para atualizar este cliente, utilize o método PUT sobre a URL /camiseta/1 e, finalmente, para apagar o cliente, utilize o método DELETE sobre a URL /camiseta/1. (COUTINHO, 2013)

Note que todas estas operações são lógicas. Isto quer dizer que utilizar o método DELETE, por exemplo, não significa necessariamente excluir o dado do banco de dados - significa apenas indisponibilizar o recurso para consumo pelo método GET. Ou seja, DELETE pode apenas marcar o dado no banco de dados como desativado. (COUTINHO, 2013)

# 8 metodologia

Nesse trabalho abordamos conceitos para a construção de uma infraestrutura compatível com o BigData. Foram fundamentados e exemplificados os tipos de bancos de dados NoSQL que trabalham de forma distribuída fornecendo uma integração muito grande com o HDFS, o sistema de arquivos do Haddop.

Será necessário vasculhar a internet para encontrar grandes volumes de dados meteorológicas que sejam confiáveis e acessíveis por meio de download ou webservices.

Buscar em livros, artigos e tutoriais na internet meios de realizar a construção de uma base não estruturada a partir de dados relacionais de um modo que seja aproveitado o maior volume de informações possíveis.

Estudar como é realizada a construção de web services para a interlocução de dados de uma forma segura e com performance admissível. Depois de entendido o procedimento, o web service será desenvolvido de forma genérica para aceitar informações de vários lugares garantindo a variedade de dados.

Será efetuado um estudo sobre alguns bancos de dados NoSql oferecidos pelo mercado e identificado os prós e contras de cada um. Depois de decidido qual banco utilizar será realizada uma modelagem especial baseada nos dados que serão obtidos pelo webservice.

Elaborar uma pesquisa em forma de entrevista com perguntas específicas e aplicar em agricultores que tem problemas meteorológicos no seu dia a dia. Com base nos resultados obtidos nas entrevistas, será realizado um estudo para definir um escopo que realmente resolva completamente ou parcialmente esses problemas, fazendo com que os utilizadores do software tenham maior facilidade em resolver entraves sobre essa questão.

Instruir-se sobre temas e projetos realizados com Big Data e analisar quais são os melhores softwares, algoritmos e técnicas para o seu uso. Empenhar-se para a construção de uma infraestrutura eficiente para o processamento desse grande volume de dados utilizando técnicas de processamento distribuído fornecido pelo software livre Hadoop fornecido pela Apache.

Explorar tecnologias para desenvolvimento de aplicações web como Java EE, AngularJS, C# e outras. Após a definição das tecnologias a serem utilizadas, realizar a implementação da aplicação assim como sua documentação, atendendo às expectativas dos agricultores entrevistados. Por fim, realizar a implantação da aplicação desenvolvida em um servidor web para que a aplicação fique acessível tanto para testes quanto sugestões úteis.

# 9 DESENVOLVIMENTO E RESULTADOS

## 9.1 Importando de Dados Meteorológicos do BDMEP

O BDMEP oferece uma grande massa de dados meteorológicos que podem ser consultados gratuitamente por estudantes e pesquisadores. Para ter acesso a esses dados é necessário acessar o site do BDMEP:

* http://www.inmet.gov.br/portal/index.php?r=bdmep/bdmep

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_1.jpg |
| **Figura 4 - Pagina de Cadastro do BDMEP** |
| **Fonte: (BDMEP, 2015).** |

Depois de ter completado o cadastro no BDMEP será enviado para seu e-mail as informações de acesso para que possamos usufruir dos dados de estações meteorológicas espalhadas por todo o país.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_3.jpg |
| **Figura 5 - Pagina de Login do BDMEP** |
| **Fonte: (BDMEP, 2015).** |

O BDMEP disponibiliza uma série histórica de dados que podem ser consultadas em três escalas: Horaria, Diária e Mensal. Cada uma dessas escalas fornecem uma quantidade de variáveis do tempo sendo a mais completa a mensal por ser medida em um espaço maior de tempo. Depois de realizar o login com suas credencias recebidas por e-mail, é disponibilizada as opções de pesquisa.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_4.jpg |
| **Figura 6 - Menu de Consulta do BDMEP** |
| **Fonte: (BDMEP, 2015).** |

Ao clicar em alguma das opções apresentadas é apresentado um formulário, aonde podemos escolher em qual período de tempo será realizada a pesquisa. Além do período de tempo, também podem ser informadas quais variáveis do tempo serão retornadas. Outro filtro interessante fornecido é possibilidade de escolher a região ou o estado em que serão buscados os dados.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_6.jpg |
| **Figura 7 - Pagina de Consulta Avançada do BDMEP** |
| **Fonte: (BDMEP, 2015).** |

No protótipo desenvolvido nessa monografia serão abordados somente dados do estado do Mato Grosso do Sul, como especificado na imagem acima.

Ao clicar em pesquisa é apresentada uma página onde existe um mapa informando a localização das plataformas meteorológicas. Clicando em cima de um ponto vermelho são apresentadas informações de latitude, longitude e altimetria. Logo abaixo desses dados existe um link aonde pode-se baixar os resultados da pesquisa para aquela plataforma.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_7.jpg |
| **Figura 8 - Pagina de Resultados do BDMEP** |
| **Fonte: (BDMEP, 2015).** |

Clicando no link “Baixar os dados” será disponibilizado um arquivo txt onde a primeira linha representa os nomes das variáveis do tempo e as posteriores os dados em questão.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_8.jpg |
| **Figura 9 – Dados da Consulta realizada no BDMEP** |
| **Fonte: (BDMEP, 2015).** |

Posteriormente esses dados serão transferidos para o Sistema de Arquivos do Hadoop e usados em funções de mapeamento e redução.

## 9.2 Importando Dados Meteorológicos do PROTIM

Como no BDMEP é necessário realizar um cadastro informando o motivo da importação dos dados. Com o usuário e senha em mãos é possível consultar variáveis do tempo de uma estação de superfície terrestre denominadas Synop. Para realizar a consulta é necessário informar qual o código da Synop escolhida, o mesmo pode ser obtido no próprio site do PROTIM.

Para esse projeto foram importados dados de uma estação localizada no aeroporto de Campo Grande no Mato Grosso do Sul. Os dados são disponibilizados por cada mês de um ano ou todos os meses de um ano que foi a utilizada.

|  |
| --- |
|  |
| **Figura 10 - Pagina de Consulta do PROTIM** |
| **Fonte: (BDMEP, 2015).** |

Ao consultar são disponibilizados dados da localização da estação terrestre como latitude, longitude e altimetria. Também pode-se ver diversas variáveis do tempo como temperatura do ar, humidade relativa do ar, pressão entre várias outras. No fim da página é possível realizar o download de todos os dados em um formato csv que é utilizado para a importação dos dados no Excel facilmente.

|  |
| --- |
|  |
| **Figura 11 – Dados da Consulta realizada no PROTIM** |
| **Fonte: (PROTIM, 2015).** |

## 9.3 Manipulando Arquivos no HDFS

O Hadoop Distributed File System é o sistema de arquivos distribuído nativo do Hadoop, que é capaz de armazenar e transmitir grandes massas de dados em maquinas de baixo custo.

Para manipular os arquivos existentes no HDFS é usada a linha de comando no servidor onde o Hadoop está rodando. Para facilitar testes implementados nesse projeto foi desenvolvido um software utilizando a linguagem Java para executar ações no HDFS.

Para que a aplicação tenha acesso ao HDFS que está rodando em uma máquina virtual externa é necessário fazer algumas configurações. O Hadoop disponibiliza uma biblioteca chamada Hadoop Core. Veja um exemplo de implementação na imagem a baixo:

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Screenshot_1.jpg |
| **Figura 12 – Implementação da permissão de acesso ao Hadoop** |
| **Fonte: (PRÓPRIA, 2015).** |

Com a biblioteca do Hadoop no projeto é possível criar uma instancia do FileSystem que é usada para realizar qualquer operação feita com a linha de comando no servidor.

Para cada operação do HDFS foi implementada uma interface gráfica para facilitar a interação. Para transferir um arquivo é necessário preencher o diretório do arquivo, o nome que será dado para o mesmo dentro do HDFS e o caminho que o arquivo será colocado dentro do HDFS. Ao clicar em transferir, se não existir, o diretório é criado e o arquivo é escrito nesse diretório.

|  |
| --- |
|  |
| **Figura 13 - Aba de Transferência de Arquivos para o HDFS** |
| **Fonte: (PRÓPRIA, 2015).** |

Também é possível listar os arquivos ou diretórios de uma pasta. Para isso é necessário somente especificar para o software o caminho no HDFS que se deseja visualizar. Para remover um diretório só é necessário o caminho do mesmo e por fim para ler um arquivo que está no HDFS também como as outras operações só é necessário digitar o caminho.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\imagem.jpg |
| **Figura 14 - Abas para listar, remover e ler Arquivos ou Diretórios no HDFS** |
| **Fonte: (PRÓPRIA, 2015).** |

## 9.4 Testando o MapReduce com dados de Temperatura

O processamento dos dados é dividido em duas etapas, a fase de mapeamento e a fase de redução. Os dados são estruturados em pares de chave-valor que devem ser definidos pelo programador.

A entrada para a fase de mapeamento são dados meteorológicos fornecidos pelo PROTIM que estão no formato de texto já armazenado no HDFS previamente, nos quais cada linha representa um registro de variáveis do tempo em uma data e hora.

Para a chave do mapeamento deve ser escolhido uma variável que agrupara os dados que serão enviados para a redução. Para esse algoritmo foi utilizado a data da medição retirando o ano, pois serão agrupadas temperaturas de todos os anos possíveis em um mês.

A função de mapeamento lê as linhas do arquivo uma de cada vez e realiza o processo de separação dos dados necessários e envia para a saída. O código da função de mapeamento é mostrado abaixo.

|  |
| --- |
|  |
| **Figura 15 – Função de Mapeamento** |
| **Fonte: (PRÓPRIA, 2015).** |

A função de redução recebe os dados processados pela função de mapeamento organizados em chave e uma lista dos valores referentes aquela chave. Os valores recebidos são as temperaturas de um dia do mês para todos os anos analisados, para reduzir o escopo da resposta fazemos uma média aritmética entre todas as temperaturas para descobrir o valor intermediário entre todas as temperaturas daquele dia. A saída da função de redução também é em estrutura de chave-valor porem agora o valor não é uma lista e sim a média da lista recebida da função map.

|  |
| --- |
|  |
| **Figura 16 – Função de Redução** |
| **Fonte: (PRÓPRIA, 2015).** |

Para esse teste foi escolhido o mês de janeiro, que é passado na configuração de um job como parâmetro para a função de mapeamento. Para realizar a execução do MapReduce remotamente de uma aplicação Java é necessário que sejam feitas algumas configurações. Uma delas é a permissão para executar uma ação no Hadoop, quando realizamos a instalação e configuração do Hadoop em um servidor Linux, todo o processo foi feito usando um usuário específico para o Hadoop que tenha permissões de administrador no servidor.

A partir da classe “UserGroupInformation” implementada na biblioteca “hadoop-core” é possível criar um canal remoto de acesso ao hadoop usando o nome de usuário citado acima, podendo assim executar ações no Hadoop sem qualquer restrição. O Hadoop fornece muitas outras formas de autenticação para seus serviços mas a segurança não é o foco principal desse trabalho, por esse motivo não será abordada com detalhes. Veja abaixo como fica a implementação das permissões necessárias.

|  |
| --- |
|  |
| **Figura 17 – Permissões para executar ações no HDFS** |
| **Fonte: (PRÓPRIA, 2015).** |

Ao executar um Job remoto no Hadoop o mesmo precisa saber aonde estão rodando os serviços que serão utilizados como o HDFS e o JobTracker. Para informar isso ao Job criamos uma instancia da classe “JobConf” e setamos as propriedades “fs.default.name” e “mapred.job.tracker” com seus respectivos valores de IP e Porta de acesso.

Depois de configurado as propriedades de acesso ao serviço é necessário informar ao Job quem são as classes de mapeamento e redução. As classes de mapeamento e redução são empacotadas em um arquivo “.jar” e enviadas ao Hadoop através da propriedade “conf.setJar("caminho");” passando o caminho como parâmetro.

|  |
| --- |
|  |
| **Figura 18 – Propriedades para executar o MapReduce no HDFS** |
| **Fonte: (PRÓPRIA, 2015).** |

O próximo passo é executar o processo de MapReduce e obter a resposta que é armazenada em um arquivo dentro do HDFS. Ao ler esse arquivo obtemos as seguintes informações organizadas em chaves e valor onde a chave é a data sem o ano e o valor é a temperatura média para aquele dia do mês.

|  |
| --- |
| C:\Users\jrcsilva\Desktop\Sem título.jpg |
| **Figura 19 – Resultado Gerado pelo MapReduce** |
| **Fonte: (PRÓPRIA, 2015).** |

# 10 CONCLUSÃO

Nesse trabalho abordamos conceitos para a construção de uma infraestrutura compatível com o BigData. Foram fundamentados e exemplificados os tipos de bancos de dados NoSQL que trabalham de forma distribuída fornecendo uma integração muito grande com o HDFS, o sistema de arquivos do Haddop.

Conhecemos os componentes principais do framework Hadoop para o processamento do BigData, como o MapReduce e o HDFS. Também exploramos alguns softwares integrados ao Haddop que auxiliam no processamento analítico dos dados fornecendo informações úteis para as empresas.

Foi realizada uma breve introdução do que é o tipo de webservice REST que é fundamental para a busca de informações em grandes bases de dados meteorológicas relacionais existentes na internet.

Concluímos que com tecnologias poderosas e de código aberto pode ser realizada a construção de uma infraestrutura moderna e eficiente para a utilização do Big Data.

Uma das maiores dificuldades para se usar o Hadoop é a sua configuração. São muitas variáveis a serem observadas para que a infraestrutura fique funcional e muitas vezes o único ponto de referência é a documentação original, que é fornecida pela empresa Apache somente em inglês, podendo não ser tão esclarecedora.

A conexão da infraestrutura do Hadoop com uma aplicação Java externa foi um dos problemas que levaram mais tempo para ser resolvido, pelo fato de que o Hadoop ainda é uma tecnologia em crescimento, sem muitos livros e manuais publicados, e a Apache fornecer uma documentação escassa sobre o assunto.

Depois de muitos testes e pesquisas o problema foi resolvido usando uma biblioteca de funcionalidades que acessa os servidores do Hadoop utilizando o protocolo HTTP sendo possível assim iniciar tarefas remotamente e interagir com o HDFS transferindo e recuperando arquivos processados pelo MapReduce.

Este trabalho foi muito importante para ampliar meus conhecimentos e a melhor compreensão dessas novas tecnologias que estão em alta nos dias de hoje. Com os conhecimentos obtidos será mais simples a construção prática do meu projeto assim como elucidar a resolução de alguns problemas em meus objetivos.

# REFERÊNCIAS BIBLIOGRÁFICAS

ALECRIM, E. **Hadoop MapReduce**: Introdução a Big Data. Disponível em: <http://www.devmedia.com.br/hadoop-mapreduce-introducao-a-big-data/30034>. Acesso em: 25 de Maio de 2015.

ALGARTECH. **Cientista de dados**. Disponível em: <http://www.algartech.com/portugues/noticias/em-noticia/mercado/cientista-de-dados-3-habilidades-que-universidades-devem-ensinar/>. Acesso em: 25 de Maio de 2015.

ANDRADE, T. P. C. **MapReduce: Conceitos e Aplicações**. Disponível em: <www.ic.unicamp.br/~cortes/mo601/trabalho\_mo601/tiago\_cruz\_map\_reduce/relatorio.pdf>. Acesso em: 25 de Maio de 2015.

BDMEP. **Banco de Dados Meteorológicos para Ensino e Pesquisa.** Disponível em: <http://www.inmet.gov.br/projetos/rede/pesquisa/>. Acesso em: 12 de Abril de 2015.

BOAGLIO, F. **MongoDB:** Construa novas aplicações com novas tecnologias. Editora Casa do COdigo, vol. 1, nº 215, São Paulo, 2015.

BRITO, R. W. **Bancos de Dados NoSQL x SGBDs Relacionais:** Análise Comparativa. Disponível em: <http://www.infobrasil.inf.br/userfiles/27-05-S4-1-68840-Bancos%20de%20Dados%20NoSQL.pdf>. Acesso em: 25 de Maio de 2015.

CHEDE, C. **Estamos definitivamente entrando na era do Big Data**. Disponível em: <https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/estamos\_definitivamente\_entrando\_na\_era\_do\_big\_data?lang=en>. Acesso em: 25 de Maio de 2015.

COUTINHO, P. C. **Primeiros passos com os serviços REST**. Disponível em: <http://www.infoq.com/br/articles/rest-introduction>. Acesso em: 25 de Maio de 2015.

DEVMEDIA**. Big Data: Armazenamento, análise e gerenciamento**. Disponível em: <http://www.devmedia.com.br/big-data-armazenamento-analise-e-gerenciamento/30 918> Acesso em: 9 de Maio de 2015.

DEVMEDIA. **Comparando o NoSQL ao modelo relacional**. Disponível em: <http://www.devmedia.com.br/comparando-o-nosql-ao-modelo-relacional/30917>. Acesso em: 12 de Abril de 2015.

GASPAROTTO, H. M. **Hadoop MapReduce**: Introdução a Big Data. Disponível em: <http://www.devmedia.com.br/hadoop-mapreduce-introducao-a-big-data/30034>. Acesso em: 25 de Maio de 2015.

HABITZREITER, P. **Vantagens da utilização do Big Data**. Disponível em: <http://www.administradores.com.br/artigos/tecnologia/vantagens-da-utilizacao-do-big-data/79939/>. Acesso em: 25 de Maio de 2015.

HANSON, J. J. **Uma Introdução ao Hadoop Distributed File System**. Disponível em: <http://www.ibm.com/developerworks/br/library/wa-introhdfs/#icomments>. Acesso em: 25 de Maio de 2015.

IANNI, V. **Introdução aos bancos de dados NoSQL**. Disponível em: <http://www .devmedia.com.br/introducao-aos-bancos-de-dados-nosql/26044> Acesso em: 9 de Maio de 2015.

ITEXTO. **Desconstruindo NoSQL: em busca de melhores termos**. Disponível em: <http://www.itexto.net/devkico/?p=1523>. Acesso em: 25 de Maio de 2015.

JONES, M. T. **Processamento de Dados Distribuídos com Hadoop** – Parte 01. Disponível em: <http://imasters.com.br/artigo/18122/desenvolvimento/processamento\_de\_dados\_distribuidos\_com\_hadoop\_parte\_01/>. Acesso em: 25 de Maio de 2015.

KNXISAAC. **Como aproveitar as informações da web e adequar as melhores para sua empresa**. Disponível em: <http://www.konics.com.br/blog/?p=2131>. Acesso em: 25 de Maio de 2015.

KOKAY, M. C. **Banco de dados NoSQL:** Um novo paradigma. Disponível em: <http://www.devmedia.com.br/banco-de-dados-nosql-um-novo-paradigma-revista-sql-magazine-102/25918>. Acesso em: 25 de Maio de 2015.

MARTINS, C. R. L. **Portal de Tecnologia da Informação para Meteorologia**. Disponível em: <http://bancodedados.cptec.inpe.br/estatisticas/cobertura.jsp>. Aces-so em: 12 de Abril de 2015.

MAYER-SCHONBERGER, V; CUKIER, K. **Big Data:** Como Extrair Volume, Variedade, Velocidade e Valor da Avalanche de Informação Cotidiana. Editora Campus, vol. 1, São Paulo, 2013.

MICROSOFT. **Introdução ao Hadoop no HDInsight:** Processamento e análise de big data na nuvem. Disponível em: <http://azure.microsoft.com/pt-br/documentation/articles/hdinsight-hadoop-introduction/>. Acesso em: 25 de Maio de 2015.

MOURA, T. G. C. **Estudo Banco de Dados espaciais distribuído**. Disponível em: <http://www.terralab.ufop.br/dokuwiki/lib/exe/fetch.php?media=terralab:equipe:mourathiago:relatorio.pdf>. Acesso em: 25 de Maio de 2015.

PARAIZO, L. **Estudo da IBM aponta nível de uso do Big Data no Brasil**. Disponível em: <http://www.noticenter.com.br/?modulo=noticias&caderno=tecnologia&noticia=02497-estudo-da-ibm-aponta-nivel-de-uso-do-big-data-no-brasil#.VWM24EYj4bg>. Acesso em: 25 de Maio de 2015.

PERERA, S. **Considerações sobre o Banco de Dados Apache Cassandra**. Disponível em: <http://www.ibm.com/developerworks/br/library/os-apache-cassandra/>. Acesso em: 25 de Maio de 2015.

PORCELLI, A. **O que é noSQL**. Disponível em: <http://www.devmedia.com.br/o-que-e-nosql-java-magazine-86/18777>. Acesso em: 25 de Maio de 2015.

PROTIM. **Portal de Tecnologia da Informação para Meteorologia**. Disponível em: <http://bancodedados.cptec.inpe.br/estatisticas/cobertura.jsp>. Acesso em: 12 de Abril de 2015.

SADALAGE, P. J; FOWLER, M. **NoSQL Essencial:** Um Guia Conciso Para O Mundo Emergente Da Persistência Poliglota. Editora Novatec, vol. 1, nº 216, São Paulo, 2013.

SATO, P. M. **GraphDB Series:** O que é um banco de dados de grafos. Disponível em: <http://www.dev.mayogax.me/o-que-e-banco-de-dados-de-grafos>. Acesso em: 25 de Maio de 2015.

SAUDATE, A. **REST:** Construa API's inteligentes de maneira simples. Editora Casa do Código, vol. 1, nº 314, São Paulo, 2013.

SILVA, B. A. **Apache Hadoop**. Disponível em: <http://www.dcomp.sor.ufscar.br/verdi/topicosCloud/Hadoop.pdf>. Acesso em: 25 de Maio de 2015.

SOUZA, C. **Comparativo Técnico entre tecnologias de banco de dados relacional, NoSQL e NewSQL**. Disponível em: <http://www.academia.edu/6559619/Comparativo\_T%C3%A9cnico\_entre\_tecnologias\_de\_banco\_de\_dados\_relacional\_NoSQL\_e\_NewSQL>. Acesso em: 25 de Maio de 2015.

TAURION, C. **Você realmente sabe o que é Big Data?**. Disponível em: < https://www.ibm.com/developerworks/community/blogs/ctaurion/entry/voce\_realmente\_sabe\_o\_que\_e\_big\_data?lang=en>. Acesso em: 25 de Maio de 2015.

TILKOV, S. **Uma rápida Introdução ao REST**. Disponível em: <http://www.infoq.com/br/articles/rest-introduction>. Acesso em: 25 de Maio de 2015.

VARDANYAN, M. **Um olhar sobre alguns bancos de dados NoSQL**. Disponível em: <http://imasters.com.br/artigo/22034/banco-de-dados/um-olhar-sobre-alguns-bancos-de-dados-nosql/>. Acesso em: 25 de Maio de 2015.

WEISSMANN, H. L. **Entendendo NoSQL**. Disponível em: <http://www.devmedia.com.br/entendendo-nosql-revista-sql-magazine-111/27851>. Acesso em: 25 de Maio de 2015.

XAVIER, O. C. **Serviços Web Semânticos Baseados em RESTful**. Disponível em: <http://www.portal.inf.ufg.br/mestrado/sites/www.inf.ufg.br.mestrado/files/uploads/Dissertacoes/OtavioCala%C3%A7a.pdf>. Acesso em: 25 de Maio de 2015.

ZHU, J. **Data Modeling For Big Data**. Disponível em: <http://www.ca.com/ us/~/media/Files/Articles/ca-technology-exchange/data-modeling-for-big-data-zhu-wa ng.pdf>. Acesso em: 12 de Abril de 2015.

ZUPPO, D; COSTA, L. S; FERNANDES, S. **Big Data:** Estudo do Ambiente, Desafios e Analise Estratégica para o Brasil. Disponível em: <http://portal.crie.coppe.ufrj.br/portal/data/documents/storedDocuments/%7B93787CAE-E94C-45C7-992B-9403F6F40836%7D/%7B53B680A0-DBBE-48B4-AB48-9D0F77B736D1%7D/RJ24\_Projeto%20Big%20Data.pdf>. Acesso em: 25 de Maio de 2015.