**NÍCOLAS HENRIQUE VIEIRA TOLEDO**

**SEBASTIÃO BATISTA DE ANDRADE NETO**

**ESTATÍSTICAS DE REDES SOCIAIS**

**COM BIG DATA**

**UNIVERSIDADE DO VALE DO SAPUCAI**

**POUSO ALEGRE**

**2015**

**NÍCOLAS HENRIQUE VIEIRA TOLEDO**

**SEBASTIÃO BATISTA DE ANDRADE NETO**

**ESTATÍSTICAS DE REDES SOCIAIS**

**COM BIG DATA**

Projeto de pesquisa apresentado à disciplina de TCC 1 do Curso de Sistemas de Informação como requisito parcial para obtenção de créditos sob orientação do Prof. Ednardo David Segura.

**UNIVERSIDADE DO VALE DO SAPUCAI**

**POUSO ALEGRE**

**2015**

# QUADRO TEÓRICO

No quadro teórico serão descritos o histórico e os conceitos que são relacionados com o tema, bem como as tecnologias que serão aplicadas no desenvolvimento do projeto.

## *Big Data*

A história de como os dados se tornaram tão vastos, gerando o que hoje conhecemos como *Big Data*, vai muito além do atual estado estabelecido. Há nove anos atrás já houve tentativas de quantificar a taxa de crescimento no volume de dados ou o que ficou conhecido como a “explosão de informação” (PRESS, 2013).

Aqui será descrito um resumo como tudo aconteceu desde 1944, quando Freemont Rider, em sua obra “The Scholar and the Future of the Research Library”*[[1]](#footnote-1)*, estimaria que “As bibliotecas universitárias americanas dobravam o tamanho a cada dezesseis anos” (RIDER, 1944, apud PRESS, 2013). Ele especulava que em 2040, a Biblioteca de Yale teria cerca de 200.000.000 de volumes e precisaria de cerca de seis mil pessoas trabalhando no serviço de catálogo.

Hal B. Becker, em sua publicação “Can users really absorb data at today’s rates? Tomorrow’s?”[[2]](#footnote-2), de 1986, estimava que a densidade de caracteres alcançada por Guttenberg era de aproximadamente de 500 caracteres por polegada cubica. Por volta do ano 2000, as memórias RAM armazenariam 1.25X10¹¹ bytes*[[3]](#footnote-3)* por polegada cubica (PRESS, 2013).

Em 1997 Michael Cox e David Ellsworth publicaram “Application-controlled demand paging for out-of-core visualization”[[4]](#footnote-4), no qual eles começaram o artigo afirmando que o tamanho extenso dos conjuntos de dados penalizam as capacidades da memória principal, disco local e até discos remotos, gerando assim um desafio interessante para os sistemas computacionais. A esse desafio, eles deram o nome de “problema do Big Data”. De acordo com eles, quando conjuntos de dados não cabem mais na memória principal, ou quando os dados não cabem nem nos discos locais, a solução mais comum é adquirir mais recursos. (PRESS, 2013).

A publicação de 1999 “*Visually exploring gigabyte data sets in real time”[[5]](#footnote-5)*, de Steve Bryson, David Kenwright, Michael Cox, David Ellsworth e Robert Haimes, dizia que computadores poderosos são uma dádiva em diversos campos de investigação, porém em contrapartida, suas operações computacionais rápidas geram quantidades massivas de dados.

Eles também afirmavam que tempos atrás, alguns megabytes[[6]](#footnote-6) de conjuntos de dados eram considerados grandes. Entretanto, na época já eram encontrados conjuntos de dados na casa dos 300 gigabytes[[7]](#footnote-7). Porém se torna um desafio entender os resultados dos dados de computações de alto desempenho. É simplesmente difícil olhar para todos os números. Além do mais, o objetivo da computação é o conhecimento, e não números. (PRESS, 2013).

Na publicação “*How Much Information?*”[[8]](#footnote-8), de Peter Lyman e Hal R. Varian feita em 2000, foi apontado que em 1999 o mundo produziu cerca de 1.5 exabytes[[9]](#footnote-9) de informação original, ou cerca de 250 megabytes para cada homem, mulher e criança na terra. Uma vasta quantidade de informação é criada e armazenada por indivíduos (democratização dos dados), e isso cresce velozmente. Chamando essa descoberta de “A dominância do Digital”, Lyman e Varian diziam que maioria da informação textual já “nascia digital” e em poucos anos, a sentença também se tornaria verdadeira para imagens (PRESS, 2013).

Também em 2000, Francis X. Diebold apresentou um documento intitulado “*Big Data Dynamic Factor Models for Macroeconomic Measurement and Forecasting*.”[[10]](#footnote-10), onde ele afirma que “Recentemente, muita ciência boa, seja física, biológica ou social, foi forçada a confrontar, além de obter benefícios, do fenômeno “*Big Data*”” (DIEBOLD, 2000, apud PRESS, 2013).

*Big Data* refere-se como a explosão na quantidade (e as vezes na qualidade) da informação relevante disponível, em grande parte, o resultado de avanços recentes e inéditas em gravação de dados e tecnologia de armazenamento. (PRESS, 2013).

Em 2001, Doug Laney, um analista no Meta Group, publicou uma nota de pesquisa intitulada “*3D Data Management: Controlling Data Volume, Velocity, and Variety.*”[[11]](#footnote-11) Uma década depois, os “3v's” se tornaram as três dimensões que definem o *Big Data*, mesmo que nas notas de Laney o termo em si não aparece (PRESS, 2013).

Em 2007 John F Gantz, David Reinsel e outros pesquisadores lançam um livro, intitulado “*The Expanding Digital Universe: A Forecast of Worldwide Information Growth through 2010*”[[12]](#footnote-12). Nele, era afirmado que em 2006, o mundo produziu 161 exabytes de dados, além de prever que entre 2006 e 2010, a informação adicionada anualmente ao universo digital aumentaria para 988 exabytes, ou dobrando a cada 18 meses. De acordo com os lançamentos de 2010 e 2012 do mesmo estudo, a quantidade de dados digitais ultrapassou esta previsão, atingindo 1.227 exabytes em 2010, e crescendo a 2.837 exabytes em 2012 (PRESS, 2013).

Em 2008 Randal E. Bryant, Randy H. Katz, e Edward D. Lazowska publicaram “*Big-Data Computing: Creating Revolutionary Breakthroughs in Commerce, Science and Society*”[[13]](#footnote-13).

Para Bryant; Katz e Lazowska (2008, p.1 e seg.) apud Press (2013):

Assim como os motores de busca têm transformado a forma de acessar as informações, outras formas de *Big Data* podem e vão transformar as atividades das empresas, pesquisadores científicos, médicos e operações de defesa e inteligência da nossa nação[...] *Big Data* é, talvez, a maior inovação na computação na última década. Nós apenas começamos a ver o seu potencial para coletar, organizar e processar dados em todas as esferas da vida. Um modesto investimento por parte do governo federal poderia acelerar consideravelmente o seu desenvolvimento e implementação.

Em 2010 Kenneth Cukier informa em seu relatório “*Data, data everywhere*”[[14]](#footnote-14) que o mundo contém uma inimaginável quantidade de informação digital que está crescendo cada vez mais mais rapidamente, e que o efeito pode ser visto em todos os lugares. Cientistas e engenheiros de computação designaram termo para o fenômeno: “*Big Data*” (PRESS, 2013).

Na publicação de 2011 “*Big data: The next frontier for innovation, competition, and productivity*”[[15]](#footnote-15) feita por James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, e Angela Byers, foi estimado que até 2009, quase todos os setores da economia americana tinham, pelo menos, uma média de 200 terabytes[[16]](#footnote-16) de dados armazenados por empresa (o dobro do tamanho do *Data Warehouse[[17]](#footnote-17)* da varejista americana *Wal-Mart* em 1999). No total, o estudo estimou que 7,4 exabytes de novos dados foram armazenados pelas empresas e 6,8 exabytes por consumidores em 2010 (PRESS, 2013).

Na publicação de 2012 “*Critical Questions for Big Data in* Information, Communications, and Society**”**[[18]](#footnote-18), Danah Boyd e Kate Crawford definem *Big Data* como um fenômeno cultural, tecnológico e acadêmico que é formado da interação entre tecnologia (capacidade computacional e precisão algorítmica para coletar, analisar, unir e comparar dados), análise (identificação padrões, a fim de fazer reivindicações econômicas, sociais, técnicas e legais) e mitologia (crença de que grandes conjuntos de dados oferecem uma forma mais elevada de inteligência e conhecimento que podem gerar perspectivas que antes eram impossíveis, com verdade, objetividade e precisão) (PRESS, 2013).

## *Storm*

De acordo com Leibiusky, Eisbruch e Simonassi (2012), o *Storm* é um sistema distribuído, confiável e tolerante a falhas desenvolvido para processar fluxos de dados.

Sua estrutura consiste em diferentes tipos de componentes. Desta forma o trabalho é dividido entre os componentes, cada qual é responsável por uma tarefa de processamento simples e específica. O fluxo de entrada do *cluster[[19]](#footnote-19)* do *Storm* é tratado por um componente chamado *spout*.

Segundo Apache (2015), um *spout* pode ser denominado como a fonte de fluxos em uma topologia. Geralmente, eles irão ler tuplas (sequencias de objetos) de uma fonte externa, posterirormente encaminhando-os para a topologia.

Continuando com Leibiusky, Eisbruch e Simonassi (2012), o *spout* por sua vez direciona os dados para um componente denominado *bolt*, o qual irá transformá-los de alguma maneira. Um *bolt* pode tanto persistir os dados em algum tipo de armazenamento, quanto direcioná-lo para outro *bolt*.

Para melhor entendimento, pode-se imaginar o *cluster* do *Storm* como uma corrente de *bolts*, cada qual executa algum tipo de transformação nos dados expostos pelo *spout*.

Dentre algumas propriedades do *Storm*, podem ser citadas:

* Simples de programar: A complexidade é reduzida;
* Suporte a várias linguagens de programação: Maior facilidade desenvolver em linguagens baseadas na JVM[[20]](#footnote-20), no entanto qualquer linguagem é suportada desde que se implemente uma pequena biblioteca intermediária;
* Tolerante a falhas: O *cluster* do *Storm* cuida dos trabalhos, realocando as tarefas quando necessário;
* Escalável: O *Storm* pode reatribuir as tarefas para as novas máquinas, caso seja necessário;
* Confiável: Todas as mensagens são obrigatoriamente processadas ao menos uma vez. Em caso de erros, a mensagem pode ser processada novamente. Assim, tem-se a garantia de nunca perder uma mensagem;
* Rápido: Velocidade é um dos fatores-chave do *Storm*;
* Transitivo: Pode-se obter exatamente a mesma semântica de mensagens para praticamente qualquer tipo de computação.

O *Storm* torna-se uma solução eficiente, rápida, confiável e de baixo custo para operações relacionadas à *Big Data*. Grandes massas de dados tem que ser processadas em tempo real, com a garantia que nenhum dado se perca. A ferramenta possibilita este tipo de solução, através de seu design e suas funcionalidades.

## Banco de dados NoSQL

Segundo Sadalage e Fowler (2013), é uma certa ironia que o termo “NoSQL” tenha aparecido pela primeira vez no final dos anos 1990 com a designação de “banco de dados de código aberto”.

Desenvolvido por Carlo Strozzi, esse tipo de banco de dados tem como característica o armazenamento de suas tabelas como arquivos do tipo ASCII[[21]](#footnote-21). Cada uma de suas tuplas é representada por uma linha, que tem os seus campos separados por guias.

O nome “NoSQL” vem do fato de não ser usado SQL[[22]](#footnote-22) como uma linguagem de consulta aos seus registros. Ao invés disso, o banco de dados NoSQL é manipulado através de *shell scripts[[23]](#footnote-23)* que podem ser combinados nos *pipelines* do UNIX[[24]](#footnote-24).

O uso do NoSQL que é reconhecido em dias atuais, remonta a um encontro que ocorreu em 11 de junho de 2009 em São Francisco, EUA, organizado pelo desenvolvedor de software Johan Oskarsson. Um exemplo que foi desenvolvido usando BigTable[[25]](#footnote-25) e Dynamo[[26]](#footnote-26) havia inspirado diversos projetos a experimentar um armazenamento de dados alternativo, e discussões a respeito do NoSQL haviam se tornado uma característica das melhores conferências de software em torno desse tempo.

De acordo DevMedia (2013), o movimento NoSQL promove soluções de armazenamento de dados não relacionais. Suas diversas ferramentas resolvem problemas relacionados a grandes volumes de dados, execução de consultas com baixa latência e modelos flexíveis de armazenamento de dados (como XML[[27]](#footnote-27) ou JSON[[28]](#footnote-28)).

O objetivo do NoSQL não é de substituir os bancos de dados relacionais, mas sim apresentar algumas soluções, que em determinados momentos, são mais adequadas e eficientes. Assim, nada se impede de se trabalhar com as tecnologias baseadas em NoSQL e os tradicionais bancos de dados relacionais em um mesmo projeto.

Os tipos de bancos de dados NoSQL são:

* Chave-Valor: Armazenam objetos indexados por chaves, e possibilitam sua busca a partir de sua respectiva chave.
* Documentos: Conjunto de documentos no formato JSON. Os documentos são tratados como objetos únicos, os quais possuem campos com os respectivos valores.
* Coluna: Formado por colunas que contem um conjunto de informações, semelhante a uma tabela.
* Grafos: Formado por dados distribuídos em forma de vértices e arestas, os quais possuem atributos tanto nas arestas quanto nos vértices.

Os bancos de dados NoSQL são implantados quando os bancos de dados relacionais já não estão suportando a demanda de dados, ou seja, estão apresentando lentidão em consultas e processamentos. Um NoSQL pode substituir um banco de dados relacional por completo, ou simplesmente, ser implantado como um “suporte”, realizando as consultas/processamentos em um grande volume de dados onde a velocidade é necessária. O tipo de NoSQL é escolhido após uma análise crítica do gestor de TI da organização, dependendo da necessidade da mesma.

### MongoDB

Segundo Chodorow (2013), o MongoDB é um banco de dados NoSQL poderoso, flexível e escalável. Ele combina a capacidade de dimensionmento com características tais como índices secundários, consultas por abrangência, classificação,  
agregações e índices geoespaciais.

O MongoDB não é um banco de dados relacional, mas sim um banco de dados orientado a documentos. Uma das principais razões para a migração dos tradicionais modelos relacionais é a necessidade de fazer um escalonamento de modo mais fácil.

Um banco de dados orientado a documentos substitui o tradicional conceito de uma "linha" com um modelo mais flexível, o "documento". Ao permitir a incorporação de documentos e *arrays*[[29]](#footnote-29), o banco de dados orientado a documento permite representar relações hierárquicas mais complexas com um único registro. Isso se encaixa naturalmente a forma como os desenvolvedores de linguagens orientadas a objetos pensam sobre seus dados.

O MongoDB foi projetado para ser escalonável. O seu modelo de dadaos orientado a documentos o permite separar os dados entre vários servidores de maneira mais fá cil. Ele se encarrega automaticamente de balancear a carga de dados através de um *cluster*[[30]](#footnote-30). Assim, ele consegue redistribuir documentos automaticamente e rotear as solicitações dos usuários para as máquinas corretas. Esta característica permite que os desenvolvedores se concentrem no desenvolvimento da aplicação, não do escalonamento.

Caso um cluster necessite de mais capacidade, novas máquinas podem ser adicionadas. Automaticamente, o MongoDB irá descobrir como os dados devem ser distribuídos entre elas.

## *Cloud Computing*

De acordo com Marinescu (2013), o ideal que a computação pode ser livremente distribuída como uma utilidade pública, do mesmo modo que a água e a eletricidade, foi formulado na década de 1960 pelo cientista da computação e visionário John McCarthy, o mesmo que defendeu a lógica matemática na inteligência artificial. Cerca de quatro décadas depois, com a internet já consolidada, a causa da “utilidade computacional” foi resgatada por grandes empresas de TI, tais como Amazon, Apple, Google, HP, IBM, Microsoft e Oracle.

Ainda de acordo com Marinescu (2013), o movimento da computação em nuvem foi um movimento iniciado em algum período em meados da primeira década do novo milênio, cuja ideia principal é que o processamento de informações pode ser feito com mais eficiência em métodos de armazenamento via internet, sendo assim alocados em um “*server farm*”, ou seja, um conjunto de servidores com capacidades computacionais extremamente superiores a de um computador convencional.

Uma das principais vantagens da computação em nuvem é que ela oferece serviços computacionais e de armazenamento escaláveis e elásticos. Todos os recursos utilizados para estes serviços são medidos. Com isso, os usuários serão somente cobrados pelos serviços que eles consomem. Aplicações científicas e de engenharia, mineração de dados, financiamento computacional, jogos, redes sociais, e muitas outras atividades computacionais de uso intensivo de dados podem se beneficiar da computação em nuvem.

A arquitetura da computação em nuvem oferece três tipos de serviços: Software como serviço (SaaS), Plataforma como serviço (PaaS) e Infraestrutura como Serviço (IaaS). Os serviços baseados em SaaS são mais designados ao usuário final. Os serviços baseados em PaaS são mais voltados a área de desenvolvimento, oferecendo assim poderosas ferramentas ao desenvolvedor. Já os serviços baseados em IaaS oferecem serviços de infraestrutura de TI completos, liberando assim a organização para focar nos negócios.

A computação em nuvem já é uma realidade empresarial em dias atuais. O número de organizações que estão adotando o paradigma é crescente.

## HTML

Segundo W3C (2015), a WWW (*World Wide Web*) foi criada em 1989, por Tim Berners-Lee no CERN (*European Organization for Nuclear Research[[31]](#footnote-31)*) em Genebra, na Suíça. Berners-Lee estava trabalhando em uma seção de serviços de computação no CERN, quando ele teve a ideia de reunir e organizar as pesquisas de diversos cientistas do mundo todo. Mais do que somente reunir, ele propôs que os textos poderiam ter *links[[32]](#footnote-32)* entre si, ou seja, durante a leitura de uma pesquisa, seria possível rapidamente exibir parte de outro artigo que contém texto ou diagramas diretamente relevantes.

Como estava tudo muito recente, era necessário ter uma implementação relativamente simples. Seguindo esse conceito, Berners-Lee desenvolveu o protocolo HTTP (*HyperText Transfer Protocol[[33]](#footnote-33)*). Desse modo, o formato de texto escrito para o HTTP foi nomeado de HTML.

Entre 1993 e 1997, o HTML iria passar pelas versões HTML+, HTML 2.0, HTML 3.0, HTML 4.0 e HTML 4.0.1, todas abrangendo diversas propostas para enriquecer as possibilidades da linguagem. Porém, em 1998, a W3C informou que o HTML não conseguiria avançar mais em suas especificações, pois se especulava que o futuro da Web era o XML. Com isso, o HTML ficou congelado na versão 4.0.1, e uma nova especificação, a qual foi chamada de XHTML foi liberada.

Ainda de acordo com W3C[[34]](#footnote-34) (2015), a ideia que a evolução do HTML podia ser reaberta foi testada em um workshop[[35]](#footnote-35) da W3C em 2004. Alguns princípios que fundamentaram o trabalho do HTML5, bem como o projeto de proposta abrangendo apenas características relacionadas a formulários, foram apresentados ao W3C pela Mozilla em conjunto com a Opera. A proposta foi rejeitada com o fundamento de que a mesma conflitava com a direção previamente escolhida para a evolução da Web. Assim, a equipe da W3C votou para continuar a desenvolver substitutos baseados em XML.

Pouco tempo depois, a Apple, Mozilla e a Opera anunciaram sua intenção de continuar trabalhando nesse esforço, formando assim o WHATWG (*Web Hypertext Application Technology Working Group[[36]](#footnote-36)*).

Em 2006, a W3C manifestou o interesse de participar no desenvolvimento do HTML5, e, em 2007, formou uma equipe para trabalhar com o WHATWG no desenvolvimento da especificação do HTML5.

Por alguns anos equipes trabalharam juntas, porém em 2011, os grupos chegaram à conclusão de que eles tinham objetivos diferentes. Em meados de 2012, uma nova equipe de edição foi introduzida na W3C para cuidar da criação de uma recomendação do HTML5 e preparar um projeto de trabalho para uma próxima versão do HTML.

O HTML5 tem como uma de suas principais funções a facilidade de manipulação do elemento. Com isso, o desenvolvedor pode modificar as características dos objetos de forma não intrusiva e transparente para o usuário final. Contrariamente as suas versões anteriores, o HTML5 fornece um número de ferramentas para o CSS[[37]](#footnote-37) e o *JavaScript*[[38]](#footnote-38) executarem suas funções de maneira mais eficiente possível.

Continuando com W3C (2015), HTML é uma abreviação de *Hypertext* *Markup* *Language* - Linguagem de Marcação de Hypertexto. A função do HTML é a publicação de conteúdo (texto, imagem, vídeo, áudio e etc) na Web.

## CSS

Segundo Pereira (2009), a função principal do HTML quando ele nasceu era apenas interligar instituições de pesquisa próximas, além de compartilhar documentos com maior facilidade. Como o foco do HTML não era a formatação de texto, à medida que ele ia se popularizando e evoluindo foram criadas funcionalidades para definir as aparências da página. Com isso, a linguagem ficava cada vez mais complexa, ficando assim mais difícil de entender. Porém o maior problema era a diferença de compatibilidade dos navegadores.

A partir destas complicações, nasceu o CSS. A principal função do CSS é separar o conteúdo e formato de um documento de sua apresentação, incluindo elementos como cores, formatos de fontes e layout, além de ser responsável por definir animações e transições. Com isso, foi proporcionada uma maior flexibilidade e controle na especificação de como as características serão exibidas, além de permitir um compartilhamento de formato e reduzir a repetição no conteúdo estrutural de uma página.

### *Bootstrap*

De acordo com Magno (2012), o *Bootstrap* é um *framework[[39]](#footnote-39)* *front-end*[[40]](#footnote-40) intuitivo criado para facilitar o desenvolvimento dos elementos de interface em páginas Web. Pode ser também utilizado como um guia para reproduzir de forma consistente os padrões de desenvolvimento consolidados pelo Twitter[[41]](#footnote-41). O *Bootstrap* também pode ser utilizado para facilitar na padronização e nas melhores práticas de desenvolvimento HTML/CSS e JavaScript, tanto para iniciantes, quanto para desenvolvedores com habilidades mais avançadas que desejam dar um passo além em interações mais complexas.

## *JavaScript*

De acordo com Powers (2009), a linguagem *JavaScript* foi inicialmente destinada a ser uma interface *script* entre uma página Web carregada no cliente do navegador (na época, o Netscape) e a aplicação no servidor. Desde a sua introdução em 1995, *JavaScript* se tornou um componente chave do desenvolvimento Web, além da descoberta seu uso em outros lugares.

Segundo Microsoft (2015), a linguagem *JavaScript* é uma linguagem de *script* interpretada com base em objetos. Sua sintaxe é semelhante a sintaxe da linguagem C.

*JavaScript* é uma linguagem fracamente tipada, ou seja, os tipos de dados das variáveis não são declarados de forma explícita. Na maioria das vezes, o *JavaScript* executa conversões de tipo das varáveis automaticamente.

# REFERÊNCIAS

APACHE. **Apache Storm**. Disponível em: <https://storm.apache.org/documentation/Concepts.html>. Acessado em 06/03/2015

CHORODOW, K. **MongoDB The Definitive Guide**. 2. Ed. Sebastopol: O’Reilly, 2013.

DEVMEDIA, **O que é NoSQL?**. Disponível em: <http://www.devmedia.com.br/o-que-e-nosql-java-magazine-86/18777>. Acessado em 03/03/2015.

LEIBIUSKY, J.; EISBRUCH, G.; SIMONASSI, D. **Getting Started Whith Storm.** Sebastopol: O’Reilly, 2012.

MAGNO, A. **Globo Bootstrap**. Disponível em:

<http://blog.alexandremagno.net/2012/08/globo-boostrap/>. Acessado em 19/04/2015.

MARINESCU, D. C. **Cloud Computing Theory And Pratice.** Waltham: Elsevier Inc., 2013.

MICROSOFT, **Java Script princípios básicos**. Disponível em: <https://msdn.microsoft.com/pt-br/library/ie/6974wx4d%28v=vs.94%29.aspx>. Acessado em 21/02/2015.

PEREIRA, A. P. **O que é CSS?**.Disponível em: <http://www.tecmundo.com.br/programacao/2705-o-que-e-css-.htm>. Acessado em 07/03/2015.

POWERS, S. **Learning JavaScript**. 2. Ed. Sebastopol: O’Reilly, 2009.

PRESS, G. **A Very Short History Of Big Data**. Disponível em: <http://www.forbes.com/sites/gilpress/2013/05/09/a-very-short-history-of-big-data/>. Acessado em 04/02/2015.

SADALAGE, P. J; FOWLER, M. **NoSQL Distilled:** A Brief Guide to the Emerging World of Polyglot Persistence**.** Crawfordsville: Pearson Education Inc, 2013.

UOL. **História do JQuery**. Disponível em <http://jqueryeter.xpg.uol.com.br/historiajquery.html>. Acessado em 07/03/2015

W3C. **Visão Geral do HTML5**. Disponível em: <http://www.w3c.br/cursos/html5/conteudo/capitulo1.html>. Acessado em 21/02/2015.

W3C. **A vocabulary and associated APIs for HTML and XHTML**. Disponível em: <http://www.w3.org/TR/html/introduction.html#introduction>. Acessado em 19/04/2015.

W3C. **A history of HTML**. Disponível em: <http://www.w3.org/People/Raggett/book4/ch02.html>. Acessado em 26/04/2015.

1. A escola e o futuro da biblioteca de pesquisa. [↑](#footnote-ref-1)
2. Os usuários podem realmente absorver dados nas taxas de hoje? Amanhã? [↑](#footnote-ref-2)
3. Byte – unidadede medida computacional. [↑](#footnote-ref-3)
4. Paginação por demanda controlada por aplicativo para visualização externa. [↑](#footnote-ref-4)
5. Explorar visualmente conjuntos de dados gigabyte em tempo real. [↑](#footnote-ref-5)
6. Megabyte - Unidadede medida computacional. [↑](#footnote-ref-6)
7. Gigabyte - Unidadede medida computacional. [↑](#footnote-ref-7)
8. Quanta informação? [↑](#footnote-ref-8)
9. Exabyte - Unidadede medida computacional. [↑](#footnote-ref-9)
10. Modelos de fator dinâmico de Big Data para medição e previsão macroeconômica. [↑](#footnote-ref-10)
11. Gestão de dados 3D: controlar o volume de dados, velocidade e variedade. [↑](#footnote-ref-11)
12. A expansão do universo digital: uma previsão de crescimento da informação Worldwide até 2010. [↑](#footnote-ref-12)
13. Computação Big-Data: Criando avanços revolucionários no comércio, ciência e sociedade. [↑](#footnote-ref-13)
14. Dados, dados em toda parte. [↑](#footnote-ref-14)
15. Big data: a próxima fronteira para a inovação, concorrência e produtividade. [↑](#footnote-ref-15)
16. Terabyte – Unidade de medida computacional. [↑](#footnote-ref-16)
17. Termo para definir o conceito de onde ocorre o armazenamento em um banco de dados de informações relativas às atividades de uma organização. [↑](#footnote-ref-17)
18. Questões críticas para Big Data em Informação, Comunicação e Sociedade. [↑](#footnote-ref-18)
19. Relacionamento entre dois ou mais computadores para que estes trabalhem de maneira conjunta no intuito de processar uma tarefa. [↑](#footnote-ref-19)
20. *Java Virtual Machine* - Máquina Virtual Java. [↑](#footnote-ref-20)
21. *American Standard Code for Information Interchange* - Codificação usada para representar textos em computadores, equipamentos de comunicação, entre outros dispositivos que trabalham com texto. [↑](#footnote-ref-21)
22. *Simple Query Language*, linguagem de banco de dados . [↑](#footnote-ref-22)
23. Linguagem baseada em *scripts*. [↑](#footnote-ref-23)
24. Sistema operacional. [↑](#footnote-ref-24)
25. Banco de dados criado pelo Google. [↑](#footnote-ref-25)
26. Serviço de banco de dados NoSQL. [↑](#footnote-ref-26)
27. *Extensible Markup Linguage* - Linguagem de Marcação Extensiva. [↑](#footnote-ref-27)
28. *JavaScript Object Notation* - Notação de Objetos *JavaScript*. [↑](#footnote-ref-28)
29. Termo da computação que significa uma lista que armazena uma coleção de elementos. [↑](#footnote-ref-29)
30. Sistema que relaciona dois ou mais computadores para que estes trabalhem de maneira conjunta no intuito de processar uma tarefa. [↑](#footnote-ref-30)
31. Organização europeia de pesquisas nucleares. [↑](#footnote-ref-31)
32. Ligações, referências. [↑](#footnote-ref-32)
33. Protocolo de transferência de hipertexto. [↑](#footnote-ref-33)
34. *World Wide Web Consortium*, consórcio internacional no qual organizações filiadas, uma equipe em tempo integral e o público trabalham juntos para desenvolver padrões para a Web. [↑](#footnote-ref-34)
35. **Reunião** de um grupo de pessoas interessados em um determinado assunto, uma atividade para **discussão** sobre um tema que é de interesse para todos. [↑](#footnote-ref-35)
36. Grupo de trabalho das tecnologias de aplicação Web hipertexto. [↑](#footnote-ref-36)
37. *Cascading Style Sheets*, formata a informação entregue pelo HTML [↑](#footnote-ref-37)
38. Linguagem de programação interpretada pelo navegador. [↑](#footnote-ref-38)
39. Abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. [↑](#footnote-ref-39)
40. Toda parte incluída na apresentação visual de um site. [↑](#footnote-ref-40)
41. Rede Social. [↑](#footnote-ref-41)