FUNDAÇÃO DE ENSINO "EURÍPIDES SOARES DA ROCHA"
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA – UNIVEM
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

Luan Silveira Pontolio

Plataforma de Extração e Recuperação de Dados na Web no Contexto de Big Data

MARÍLIA
2014

Luan Silveira Pontolio

Plataforma de Extração e Recuperação de Dados na Web no Contexto de Big Data

Trabalho de Curso apresentado ao Curso de Bacharelado em Sistemas de Informação da Fundação de Ensino "Eurípides Soares da Rocha", mantenedora do Centro Universitário Eurípides de Marília – UNIVEM, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador
Profº. Dr. Elvis Fusco

MARÍLIA
2014

CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Luan Silveira Pontolio

Plataforma de extração e recuperação de dados na Web no contexto de Big Data

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Sistemas de Informação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Sistemas de Informação.

Nota: 10 (DEZ)

Orientador: Elvis Fusco

1°. Examinador: Leonardo Castro Botega

2°. Examinador: Ricardo José Sabatine Ricardo Sabatine

Marília, 03 de dezembro de 2014.

Silveira Pontolio, Luan

Plataforma de Extração e Recuperação de Dados na Web no Contexto de BigData / Luan Silveira Pontolio; orientador: Profº. Dr. Elvis Fusco. Marília, SP: [s.n.], 2014.

75 folhas

Monografia (Bacharelado em Sistemas de Informação): Centro Universitário Eurípides de Marília.

Dedico este trabalho à toda a minha família, que apoiaram-me em todos esses anos.

AGRADECIMENTOS

Agradeço a Deus, minha família e amigos, que estiveram sempre ao meu lado apoiando-me em todos os momentos alegres e difíceis, dando-me forças para que conquista-se meus sonhos e ambições.

Ao meu orientador Profº. Dr. Elvis Fusco que acreditou em meu potencial e guiou-me nesta jornada de conclusão de curso, cumprindo muito mais que seu papel de professor e orientador, tendo assim meus eternos agradecimentos.

Aos professores Ricardo José Sabatine e Giuliana Marega Marques pelos conselhos e ajuda ao longo do trabalho de conclusão de curso.

Ao Centro Universitário Eurípides de Marília e seu corpo docente, que garantiram ao longo desses quatro anos o meu crescimento profissional e acadêmico.

A empresa Boa Vista Serviços que esteve comigo neste último ano, e que ofereceu a oportunidade de atuar na área.

E a todos que direta ou indiretamente fizeram parte da minha formação, o meu muito obrigado.

“O futuro pertence àqueles que se preparam hoje para ele.”
(Malcom X)

RESUMO

Com a dispersão de dados de interesse para empresas e organizações em diversos domínios na Web e em formatos distintos, torna-se cada vez mais necessário a capacidade de obtê-los e para isso é preciso oferecer maneiras de extrair esses dados, de modo a garantir a sua confiabilidade para o armazenamento correto. Técnicas de extração de dados, em especial Web Scraping (robô de busca), permitem a captação de tais dados. Neste contexto este trabalho visa o estudo de técnicas de extração de dados, tendo como base o domínio Web, e por meio deste, concretizado no desenvolvimento de uma plataforma que ofereça a capacidade de extrair essas informações por meio da parametrização de robôs de busca, permitindo ao usuário a autonomia de sua criação.

Palavras-Chave: Big Data, Extração de Dados, Web Scraping.

ABSTRACT

The dispersion of interest data to businesses and organizations in several domains on the Web, and in different formats, it becomes increasingly necessary the ability to get them and for that is needed to provide manners to extract these data, to ensure its reliability for the correct storage. Techniques of data extractions, in particular Web Scraping (search robot), allows the capture of such data. This project aims to study techniques for data extraction, based on the Web domain, and through this, it materializes in the development of a platform that offers the ability to extract this information by means of parameterization of search robots, allowing the user autonomy of its creation.

Keywords: Big Data, Data Extraction, Web Scraping

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

W3C	WORD WIDE WEB CONSORTIMUM
XML	EXTENSIBLE MARKUP LANGUAGE
HTML	HIPER TEXT MARKUO LANGUAGE
CAPTCHA	COMPLETELY AUTOMATED PUBLIC TURING TEST TO TELL COMPUTERS AND HUMANS APART
IP	INTERNET PROTOCOL
NOSQL	NOT ONLY SQL
SQL	STRUCTURED QUERY LANGUAGE
API	APLICATION PROGRAMMING INTERFACE
IBM	INTERNATIONAL BUSINESS MACHINES
IDC	INTERNATION DATA CORPORATION
E.U.A	ESTADOS UNIDOS DA AMERICA
TDWI	THE DATA WAREHOUSING INSTITUTE
DBMS	DATABASE MANAGEMENT SYSTEMS
JSON	JAVASCRIPT OBJECT NOTATION
SGBD	SISTEMA GERENCIADOR DE BANCO DE DADOS
SGBDR	SISTEMA GERENCIADOR DE BANCO DE DADOS RELACIONAIS
ACID	ATOMICIDADE, CONSISTÊNCIA, ISOLAMNETO E DURABILIDADE
CAP	CONSISTENCY, AVAILABILITY E PARTTION TOLERANCE
RAM	RANDOM ACCESS MEMORY
POSIX	PORTABLE OPERATION SYSTEM INTERFACE
SOA	SERVICE-ORIENED ARCHITETURE

WWW	WORD WIDE WEB
HTTP	HYPERTEXT TRANSFER PROTOCOL
RSS	REALY SIMPLE SYNDICATION
SMS	SHORT MESSAGE SERVICE
URL	UNIFORM RESOURCE LOCATOR
REST	REPRESENTATION STATE TRANSFER
CSS	CASCADING STYLE SHEETS
ORM	OBJECT-RELATIONAL MAPPING
IA	INTELIGENCIA ARTIFICIAL

SUMÁRIO

INTRODUÇÃO

Em 1994, quando Tim Bernes-Lee fundou a Word Wide Web Consortium (W3C), organização dedicada a desenvolver tecnologias interoperáveis não-proprietárias para Web, tornou-a universal e acessível a todos, sendo possível com a criação de padrões chamados de *recommendations* (ou recomendações), que incluíam Extensible Markup Language (XML) e o Hyper Text Markup Language (HTML, que agora encontra-se em sua quinta versão, o HTML5), entre várias outras tecnologias que apoiaram o crescimento deste padrão, possibilitando que inúmeras fontes de dados fossem inseridas neste novo ambiente (DEITEL et al., 2003).

A Internet possui hoje muitos dados de relevância, disponíveis em documentos Web, porém seu modelo de publicação das informações permite aos seus usuários um modo informal de publicá-las. Pois tais informações são apresentadas em diversos formatos e contextos, podendo ser representados de forma estruturada, semiestruturada e não estruturada.

A geração de dados se deu por meio de diversos domínios e estão na ordem de algumas dezenas ou centenas, de *terabytes*, os quais, cerca de 2,5 quintilhões de *bytes* existentes hoje, 90% dos mesmos foram gerados somente nos últimos dois anos (KAKHANI M., KAKHANI S., BIRADAR; 2013). Com esta imensa quantidade de dados existentes, novos grandes desafios surgem na forma de recuperação, armazenamento e processamento de consultas em várias áreas da computação, e em especial na área de bases de dados, mineração de dados e recuperação da informação.

Nos processos de publicação das informações no domínio Web, o seu aumento é exponencial, o qual oferece ao mesmo tempo a necessidade de instaurar formas ou meios de obtê-las, pois sua análise e contextualização permite o estabelecimento de novos nichos de negócio, previsões futuras com base nos acontecimentos passados, bem como a estruturação de organogramas relacionais sobre toda e qualquer perspectiva de dados disposta em redes sócias, blogs ou fóruns.

A área de extração de dados concede as técnicas necessárias para a captura de dados dispostos nestes domínios, pela utilização de diversas técnicas para efetuar esta ação, e algumas sobressaem-se em relação as outras, como Web Scrapings, robôs de busca, que possuem componentes de inteligência que simulam a navegação Web humana entre as páginas, percorrendo as suas estruturas HTML e permitindo que o processo de busca, captação e extração seja realizada de forma eficaz e consistente de certa forma (MASNICK; 2009).

Como as informações podem apresentar-se em diversos formatos no domínio Web, estabelecer conceitos para que a extração dos dados seja eficiente é uma tarefa muito complexa, pois as estruturas (como o HTML) em que os dados estão inseridos, sofrem com mudanças constantes, as quais são necessárias que os mecanismos de busca sejam alterados da mesma forma.

Os mecanismos de busca, realizam a extração de dados por meio das estruturas preestabelecidas nos domínios, quando à ocorrência de mudanças que tem por objetivo impedir que tais aplicações realizem a extração das informações, como por exemplo validações de Captchas, JavaScript e bloqueios de endereços IPs, faz-se necessário que os mesmos tenham a capacidade de identificar e posteriormente solucionar estes problemas, permitindo que sejam obtidos os dados.

A extração de dados permite a captura dos dados, mas a capacidade de persistir estes grandes volumes concretizou-se por meio da criação dos modelos de bancos de dados não relacionais, ou simplesmente, “Not Only SQL” (NoSQL), o qual forneceu perspectivas diferentes as que foram geradas nos modelos tradicionais baseados em interfaces Structured Query Language (SQL), permitindo que este trabalho seja realizado suportando a manipulação de diversos modelos de dados (WYLIE et al.; 2012).

Iniciou-se assim a era do Big Data, que trouxe ao mercado a necessidade de um processo de análise de dados e inteligência analítica acoplado às estratégias de negócios. Com isso, diversas áreas podem auxiliar esta demanda crescente, como é o caso da extração e recuperação de dados, responsável pela obtenção de informações referente a um determinado domínio ou vários.

Neste trabalho o domínio de informação é a Web, por possuir dados de relevância para organizações e usuários, em especial especialistas na área de Tecnologia de Informação, que utilizam tais recursos para a realização de análise e armazenamento em grandes bases de dados, visando a obtenção de insumos necessários para a geração de conhecimentos sobre seus negócios. Para isso foi estabelecida uma plataforma de extração de dados baseada em serviços Web, que correspondem respectivamente a um serviço de manipulação das regras estabelecidas para a extração de dados, e um outro serviço capaz de realizar tal tarefa em conformidade com envio de tais regras.

Motivação e Justificativa

A Internet disponibiliza diversos dados no formato Big Data, que podem ser de grande valia para empresas ou organizações, a sua importância aumenta com o estudo e aprimoramento de ferramentas e técnicas de extração, análise e organização de dados. Algumas técnicas, como Web Scraping, são utilizadas para a realização da extração de dados de páginas no contexto Web. Assim torna-se essencial a sua utilização neste trabalho, a fim de fornecer a maior confiabilidade a extração.

Objetivos Gerais e Específicos

Como objetivo geral, esta pesquisa visa o estudo e demonstração de métodos e técnicas referentes a área de extração de dados pertinentes no ambiente Web, oferecendo insumos necessários para a construção de serviços e tecnologias deste cenário, focados na área apresentada. Sendo assim, como desenvolvimento deste trabalho é proposto a construção de uma API utilizando o estilo de arquitetura RESTful, capaz de capturar as informações das páginas por meio da parametrização de um Web Scraping, e a criação também de um serviço fornecedor e consumidor das informações necessárias para a realização da extração de dados, voltados para o auxílio de profissionais dessa área.

As especificações realizadas ao longo do desenvolvimento deste trabalho podem ser observadas em sequência, como:

1. A generalização de métodos de extração de dados, utilizando-se um robô de busca Web Scraping, que possua seus métodos parametrizados os quais dependem do estabelecimento das regras de extração por parte de seus usuários, a fim de fornecer uma maior variedade de páginas Web para a captação dos dados do tipo Big Data;
2. O desenvolvimento de uma Aplication Programming Interface (API), que abstrai um Web Scraping, como um serviço Web, baseado nas características típicas da arquitetura RESTful, a fim de fornecer a especialista da área de dados uma ferramenta de captação de dados Webs;
3. O desenvolvimento de um outro serviço com características que especifiquem a utilização da API, por meio da criação das regras do agente de busca;
4. A demonstração da utilização do MongoDB no processo de desenvolvimento do serviço Web Cliente; e

5. A finalização do desenvolvimento das duas aplicações com o estabelecimento de uma plataforma de extração e recuperação de dados na Web no contexto de Big Data.

Organização do Trabalho

O primeiro capítulo faz referência aos conceitos de Big Data e quais fatores deram início a ele, relatando as expectativas e desafios em relação ao aumento da massa de dados em diversos domínios, como a Web, proporcionando novas formas de enxergar a utilização dos dados, por meio de suas tecnologias.

O segundo capítulo tem por objetivo apresentar o modelo de banco de dados não relacional (NoSQL), o qual trouxe novas perspectivas na manipulação de dados de diversos formatos, como estruturados, semiestruturados e não estruturados. Além de caracterizar sua importância diante do cenário de Big Data.

O terceiro capítulo referencial, transcreve os conceitos, desafios e tecnologias relacionadas a área de extração de dados, na Web, que oferece os meios necessários para a captura dos mesmos.

O quarto capítulo é contextualização das metodologias utilizadas durante o período de pesquisas e desenvolvimento da pesquisa, apresentando toda a arquitetura planejada e como a mesma foi implementada com base nos conceitos e tecnologias relatados.

O quinto capítulo demostra os resultados obtidos com a consolidação da plataforma desenvolvida e como a pesquisa concretizada trouxe uma alternativa para realizar a captação de dados no ambiente Web.

A conclusão faz referência a toda pesquisa apresentado finalizando um ciclo das atividades e também conta com possíveis trabalhos futuros que podem agregar ainda mais para a área e suas tecnologias.

1 BIG DATA

As ferramentas para o processamento de dados tornaram-se essenciais na constituição de novos paradigmas para elaboração de estratégias de negócios, entre elas estão algumas como Business Intelligence (B.I), Data Mining (ou Mineração de Dados) e Data Wherehouse (vulgo Armazém de Dados). Porém essas tecnologias não suportam o grande volume de dados gerados em diversos ambientes e em inúmeros formatos, neste cenário novas tecnologias surgem com a expectativa de suprirem esses desafios.

Com os avanços tecnológicos da última década surgiram novas formas para a coleta e análise de dados em diversos domínios, como da Internet. Organizações como a Amazon, Google, Facebook, entre outras, demonstraram como grandes quantidades de dados, podem e devem ser utilizadas, para que seja realizada uma melhor analise sobre determinados conjuntos de dados, fornecendo ao fim informações concisas e precisas sobre os mesmos (VAILAYA; 2012).

O termo Big Data, referem-se ao aumento no conjunto de dados em diversos cenários, e que, tornaram-se difíceis de trabalhar com o uso de ferramentas de gerenciamento de banco de dados tradicionais. As implicações típicas incluem a captura, armazenamento, pesquisa, compartilhamento, análise e visualização. Os benefícios de trabalhar com conjuntos de dados cada vez maiores, é permitir que profissionais da área de dados possam discernir e validar com maior precisão e exatidão (COURTNEY; 2012)

1.1 Conceito de Big Data

A expansão da Internet facilitou a integração de vários módulos de comunicação, ampliando exponencialmente o acesso às informações e transformando qualquer pessoa em produtora de seu próprio conteúdo. Iniciou-se assim a era do Big Data, que trouxe ao mercado a necessidade de um processo de análise de dados e inteligência analítica acoplado às estratégias de negócios.

A quantidade de informações que são geradas em diversos domínios como por exemplo, Web, redes de sensores e diversos outros meios de comunicação, estão na ordem de algumas dezenas ou centenas, de *terabytes.* Com esta imensa quantidade de dados gerados, novos grandes desafios surgem na forma de manipulação, armazenamento e processamento de

consultas em várias áreas de computação, e em especial na área de bases de dados, mineração de dados e recuperação de informação (VIERA et al; 2012).

A crescente demanda por espaço de armazenamento nos últimos anos, acarretam na ampliação de estruturas que suportem essas novas características. De acordo com Sigiroglu e Sinanc (2013) até 2003 foram gerados cerca de 5 *hexabytes* (1018 bytes) de dados, já em 2012 com a ampliação do mundo digital para 2,72 *zettabytes* a mesma quantidade era gerada somente em dois dias, com base nesses números é previsto que a quantidade dobre a cada dois anos, atingindo aproximadamente 8 *zettabytes* de dados em 2015.

A International Business Machines (IBM) realizou uma pesquisa em 2013, com base nas mais variadas fontes de dados, da qual foram coletados dados sobre o clima, mídias sociais, fotos digitais e vídeos, registros de transação de compras e serviços telefônicos, resultando ao final uma estimativa de que cerca de 2,5 quintilhões de *bytes* de dados eram criados por dia, e 90% dos dados existentes neste período foram criados somente nos anos de 2011 e 2012 (KAKHANI M.; KAKHANI S.; BIRADAR, 2013). Outra pesquisa realizada em 2011, da Internation Data Corporation (IDC) comprovavam um aumento ainda mais expressivo, ao apresentar que os dados globais iriam crescer 50 vezes até 2020, e informações não estruturadas - como arquivos, e-mails e vídeos - serão responsáveis por 90% de todos os dados criados durante a próxima década.

O conceito Big Data aplica-se às informações que não podem ser processadas ou analisadas de maneira tradicional utilizando processos e ferramentas convencionais (ZIKOPOULOS et. al., 2012). Cada vez mais as organizações enfrentam este crescente aumento na riqueza de dados, porém ao mesmo tempo estas informações apresentam formatos semiestruturados e também não estruturados, esses desafios agravam devido as mais variadas fontes de dados.

> Estes dados são gerados a partir de transações on-line, e-mails, vídeos, áudios, imagens, registros, publicações, pesquisas, registros de saúde, interações de redes sociais, dados científicos, sensores, telefones celulares e suas aplicações (SAGIROGLU; SINANC, 2013).

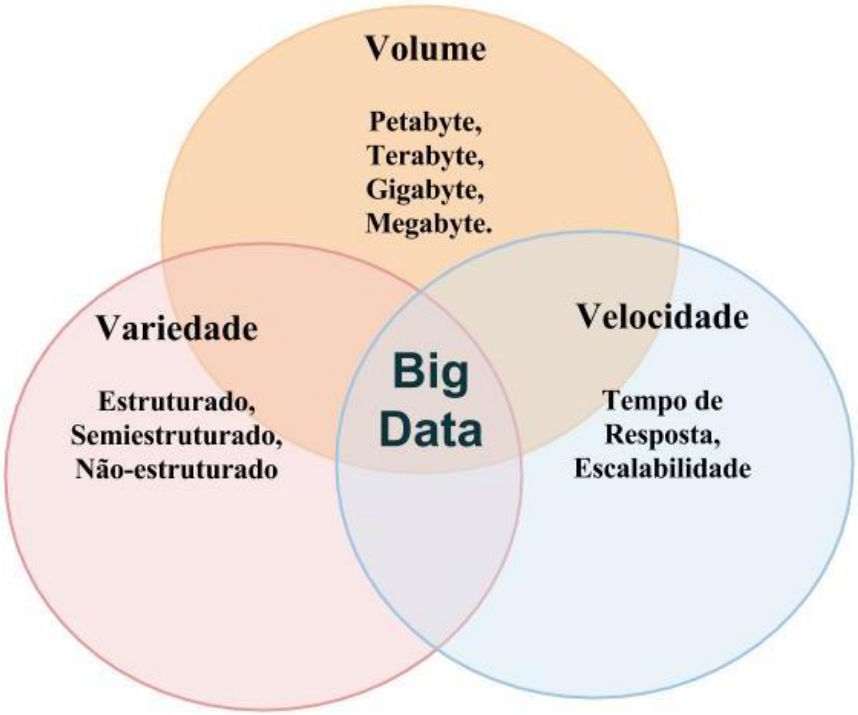
Um relatório apresentado pela Cysneiros Consultores Associados para a Secretaria de Ciência e Tecnologia do Estado de Pernambuco, desenvolvido por Cavalcanti (2013 apud ZIKOPOULOS et al., 2012), demonstra alguns pontos importantes referente ao conceito de Big Data, tais como: (1) suas soluções são ideias para analisarem dados estruturados, semiestruturados e não estruturados a partir de fontes heterogêneas de informação; (2) possibilita a analise interativa e exploratória, quando medidas de negócios com dados não são

pré-determinadas; e (3) oferece recursos tecnológicos para resolverem desafios de informações que não podem ser tratadas em banco de dados relacionais tradicionais.

As características deste novo conceito tornam-se mais evidente quando analisado suas principais propriedades, ilustrados na a Figura 1. Os autores sobre o tema Big Data definem três propriedades básicas, são elas variedade, velocidade e volume, cada qual compreende a verdadeira importância sobre o mesmo:

1. Variedade, corresponde à falta de relacionamento entre os dados, ou seja, devido a ele suportar uma grande quantidade de dados de diversas fontes, podendo ser estruturados (são dados relacionados a partir de uma base de dados), semiestruturados (são dados em que sua representação podem conter ou não uma estrutura, como em um documento XML) e não estruturados (são composto por textos, imagens, vídeos e etc.) (KAKHANI M.; KAKHANI S.; BIRADAR, 2013) (SIGROGLU; SINANC, 2013) (GERHARDT; GRINFFIN; KLEMANN, 2012);
2. Volume, Big Data supera a armazenagem de dados convencionais, como já mencionado, cerca de 2,5 *terabytes* de dados eram gerados por dia no ano 2012 (McAfee; Brynjolfsson, 2012), a quantidade atualmente ultrapassa os *terabytes* e petabytes. A grande escala e aumento de dados ultrapassam as técnicas de armazenamento e de análise tradicionais isto gera desafios, e as tecnologias referentes a este conceito visam solucioná-los (KAKHANI M.; KAKHANI S.; BIRADAR, 2013); e
3. Velocidade, para muitas aplicações, a velocidade de criação de dados é ainda mais importante do que o volume. Neste contexto a velocidade mede o tempo de criação e agregação dos dados. De acordo com Kaisler et. al (2013), a analise deve ser realizada em microssegundos, decisões devem ser tomadas a respeito dos dados capturados e possuindo relevância quando combinados com outros dados.

Figura 1 - Propriedades típicas de Big Data.



Fonte: Próprio Autor

Porém alguns autores acrescentaram duas novas propriedades, refinando o conhecimento a respeito de seu conceito, são elas, complexidade e valor, descritos na sequência:

1. Veracidade, com as enormes quantidades de dados que são obtidos para o cenário de Big Data, ocasionam em erros estatístico devido sua grande variedade, portanto garantir a qualidade e precisão das informações colhidas é essencial para as tecnologias e ferramentas deste contexto (OHLHORST; 2013) (KATAL; WAZADI; GOUDAR, 2013).
2. Valor, com base nas informações capitadas outra questão é abordada, para que possa medir se as mesmas são valiosas ou não, para a aplicação em que estão inseridas, em vista que os usuários podem executar certas consultas em relação aos dados armazenados, e portanto, podem deduzir resultados importantes a partir dos dados filtrados e obtidos, para classificá-los de acordo com as dimensões que necessitam (KATAL; WAZADI; GOUDAR, 2013).

As propriedades apresentadas refletem a grandiosidade desse novo conceito e suas tecnologias que asseguram estas propriedades, devido ao aumento nos fluxos de informações sendo criadas, transmitidas, analisadas, relacionadas, consultadas e recebidas, os cenários

convencionais estão sendo remodelados, como o que ocorre em organizações, comunidades científicas e principalmente no comportamento de pessoas. Suas tecnologias apresentam alternativas para suprirem esta demanda, e garantir a confiabilidade nos dados e seu potencial, gerando impactos transformadores necessários na solução de tarefas e atividades diárias (GERHARDT; GRIFFIN; KLEMANN; 2013).

1.2 Desafios da era Big Data

O fato de real importância sobre o tema (Big Data), é sem dúvida alguma, a sua capacidade de processar grandes quantidade de dados. Para Zikopoulos et al. (2012, p. 15), a interpretação desse termo pode ser realizada de diferentes maneiras, porém em conformidade as três principais propriedades, velocidade, volume e variedade.

Diferente do armazenamento tradicional, em que os dados precisam ser relacionados, limpos e até mesmos formatados, em um Big Data estas especificações são diferentes, como já mencionado neste capítulo (KATAL; WAZADI; GOUDAR, 2013). Em relação a este tema, um ponto chave para qualquer tecnologia desta alçada é a verificação do fluxo de dados, pois torna-se de alta complexidade controlá-los de modo a oferecer confiabilidade dos mesmos (SIGROGLU; SINANC, 2013).

Seguindo este princípio uma pesquisa realizada recentemente pelo Diretor de Pesquisas para o Gerenciamento de Dados do The Data Warehousing Institute (Instituto de Armazenagem de Dados, TDWI), Russom (2013), afirma que muitas organizações adotam novas plataformas que lhes permitem superar alguns dos problemas em relação ao desempenho do fluxo de dados, para solucionar principalmente a escalabilidade do volume de dados e o seu processamento em tempo real, oferecendo uma alternativa viável ao problema. Em sua pesquisa, é aborda uma questão de suma importância, em relação ao problemas e oportunidades no gerenciamento das tecnologias de Big Data, com base na experiência de profissionais da área de dados, ocasionando nos seguintes resultados:

1. 89% dos profissionais, consideram que suas tecnologias são ideias para a exploração de dados das mais variadas fontes e analises preditivas, para a descoberta de novos fatos sobre clientes, mercados, sócios e investidores, custos e operações;
2. 11% dos profissionais, afirmam que um Big Data apresenta desafios técnicos, devido ao seu tamanho e diversidade. Somente o volume de dados torna-se um empecilho para algumas organizações.

De acordo com Zikopoulos et al. (2012) algumas das vantagens apresentadas referente ao gerenciamento de informações utilizando tecnologias para grandes bases de dados – permitindo a compreensão do número alto de aceitação por parte de profissionais da área – é que elas não restringem-se aos conceitos tradicionais de armazenagem, tornando possível a preservação dos dados e o acesso a informações de maneira clara.

Em contra partida, na abordagem de Katal et al. (2013), sobre as soluções de Big Data que proporcionariam o estabelecimento de novas estratégias de negócios, com base na análise de imensos volumes de dados nas organizações, poderiam ser possíveis utilizando-se como base a diversos conjunto de dados, como por exemplo, alguns que são expressos a seguir: (1) Logs de armazenamento; (2) Analise a partir de dados de sensores; (3) Análise de dados de riscos; e (4) Analise em mídias sociais.

1. Muitas organizações registram grandes quantidade de dados, como *logs* de armazenamento de suas atividades, porém este armazenamento é realizado somente por um curto período de tempo, devido a utilização de sistemas tradicionais que não são capazes de lidar com seu volume de dados. A análise com Big Data não só garante a exploração sobre os dados disponíveis para identificar possíveis falhas, mas também aumenta a longevidade de seu armazenamento;
2. Devido à falta de infraestrutura de armazenamento e técnicas de análise de dados, inviabiliza a captação dos mesmos em grandes quantidades, como os disponíveis em equipamentos de transmissão de dados (como sensores);
3. Torna-se importante para as instituições financeiras a modelagem de dados, a fim de calcular os riscos e diminuí-los em níveis aceitáveis. Um grande conjunto de dados são potencialmente subutilizados e devem ser integrados dentro de modelos para determinar os padrões de risco com mais precisão; e
4. A maioria das tecnologias de Big Data estão sendo utilizadas para a análise de comportamento de clientes em mídias sociais, em relação a opiniões sobre produtos e serviços, permitindo que empresas tenham o *feedback* de seus clientes, para que melhores decisões de negócios sejam tomadas.

Algumas das grandes problemáticas hoje são as formas em que os dados estão sendo apresentados, técnicas para transformação de dados não estruturados em formatos estruturados para que posteriormente sejam inseridos em contextos, tornaram-se também grandes desafios devido a alguns aspectos específicos, como a (1) Heterogeneidade; (2)

Escala; (3) Escopo; (4) Privacidade e Segurança; e (5) Arquitetura de Sistemas, assim descritos na sequência:

1. Heterogeneidade, os dados extraídos de fontes distintas não possuem tamanho e estrutura adequada para uma primeira análise;
2. Escala, técnicas de processamento de dados paralelos não são uteis para o suporte, devido as arquiteturas apresentadas;
3. Escopo, necessário que encontre-se elementos precisos referente ao contexto dos dados, após uma análise devidamente executada;
4. Privacidade e Segurança, para que os dados possam ser utilizados é de suma importância desenvolver algoritmos que randomizem dados pessoais de usuários entre um grande conjunto de dados; e
5. Arquitetura de Sistemas, dados analisados e estudados podem ser utilizados para diversas finalidades, assim é preciso que a arquitetura do sistema primário seja flexível e capazes de executar diferentes cargas de trabalho;

A análise, organização, recuperação e modelagem de dados são desafios fundamentais devido ao aumento no fluxo de dados, forçando organizações e pessoas a discernir sobre que decisões tomar em relação a estes requisitos para a geração de informações confiáveis. Como a maioria dos dados são gerados diretamente em formatos digitais, hoje, o desafio é influenciar na criação de conteúdo facilitando a vinculação entre vários dados permitindo a criação de estruturas confiáveis para os mesmos (KALE; DANDGE, 2014).

1.3 Técnicas e Tecnologias

Os desafios causados pelo aumento de dados estimularam as pesquisas e trabalhos para que soluções precisas e confiáveis sejam utilizadas em campos acadêmicos e organizacionais, oferecendo inúmeras formas de geração e reaproveitamento de dados. Existem variedades de métodos, aplicações e ferramentas desenvolvidas para processar e analisar grandes volumes de dados (KAKHANI M.; KAKHANI S.; BIRADAR, 2013).

Algumas técnicas como Database management systems (DBMSs), Structured Query Language (SQL), segundo Russom (2013), são importantes para o gerenciamento de grandes conjuntos de dados, pois existem muitos tipos e estruturas diferentes de dados, e sua grande parcela ainda são relacionais. Outro formato de dados secundários (ou semiestruturados) aderem aos padrões de Extensible Markup Language (XML) e JavaScript Object Notation (JSON) e são utilizados para o envio de informações, na maioria das vezes entre aplicações

(como Application Programming Interface, ou, API), podendo conter nome, e-mail, informações de localidade entre outras a respeito de pessoas, que por meio de seus elementos descritivos possibilitam a identificação de seus dados para a compreensão e armazenamento adequado.

Outro aspecto que deve ser abordado é a compreensão que as tecnologias gerenciadoras de banco de dados convencionais, tais como MySQL, Oracle, DB2, PostgreSQL, SQL Server entre outros são uma parte importante e relevante de uma solução geral analítica. Na verdade, eles se tornam ainda mais vitais quando usados em conjunto com plataformas de Big Data.

A compreensão desse aspecto é delimitada em relação à quando deve ser utilizado tais tecnologias, pois os Sistemas Gerenciadores de Bancos de Dados (SGBDs) tradicionais são eficientes para se trabalhar com dados estruturados, porém como a grande maioria dos mesmos são gerados hoje de modo não convencional, ou seja, não seguem padrões de formatos e tipos (como *int*, *string*, *char* etc.) entre outras características, tornam-se incapazes de trabalhar neste contexto sozinhos.

Como já expresso no capítulo, modelos tradicionais de armazenagem não suportam volume de dados gerados, mesmo assim são importantes, mas algumas frustrações na sua utilização, que segundo Sadalage e Fowler (2013), caracterizam a diferença entre o modelo relacional e as estruturas de dados na memória, comumente chamada de incompatibilidade de impedância.

O termo incompatibilidade de impedância surgiu na década de 1990, o qual acreditava-se que os bancos de dados relacionais fossem substituídos por bancos de dados que repetissem no disco para as estruturas de dados da memória, tal década foi marcada pela criação de linguagens de programação orientada a objetos e com elas vieram os bancos de dados também orientados a objetos, ambos com a intenção de se tornarem o ambiente dominante para o desenvolvimento de software.

Com essas adversidades causadas por tais problemas, uma solução para enfrentar os desafios no contexto Big Data em relação ao volume de dados, é o movimento denominado "*Not only SQL*" (Não somente SQL, ou NoSQL), que promove diversas soluções inovadoras de armazenamento e processamento de grandes bases de dados. Suas soluções foram inicialmente criadas para sanar problemas referentes as aplicações Web, pois proporcionam arquiteturas escaláveis com grande facilidade de forma horizontal, sendo permitido fornecer mecanismo de inserção de novos dados de forma incremental e eficaz (VIERA et al.; 2012).

Este movimento não define exclusivamente os aspectos de bancos de dados, mas tem por abordar uma visão bem mais ampla e concisa ao apresentar suas características, que são elas descritas como: não-relacional, distribuído, de código aberto e escalável horizontalmente, ausência de esquema ou esquema flexível, suporte à replicação nativo e acesso via API simples. Devido aos altos graus de complexidade de dados, principalmente os que são encontrados na Web, faz-se cada vez mais necessário a cobrança de atingir-se altos graus de paralelismo, processamento de grandes volumes de dados e distribuição de sistemas em escala global (ALMEIDA, BRITO; 2010).

1.4 Considerações Finais do Capítulo

As soluções de Big Data são amplamente relacionadas a armazenamento de grandes volumes de dados, mas um ponto importante em sua concepção é a forma ou maneira de manipular as informações disponíveis em diversos domínios. Alternativas para essa ação existem desde de antes da fundamentação de seu conceito e surgiram para auxiliar no desenvolvimento de suas aplicações, possibilitando que ações de recuperação e análise de dados tornassem mais ágeis.

2 NOT ONLY SQL (NOSQL)

Os Sistemas de Gerenciamento de Banco de Dados Relacionais (SGBDR, ou somente SGBD) tradicionais, contam com alta disponibilidade de gerenciamento centralizado, permitindo que processos de validação, verificação, integridade dos dados, controle de concorrência, recuperação de falhas, segurança, controle de transações, otimização de consultas, dentre outros aspectos, garantam uma maior confiabilidade no armazenamento de dados. Tais características fizeram com que os números de seus usuários crescessem, possibilitando que aplicações mais completas e complexas fossem desenvolvidas (BRITO; 2010).

No cenário atual em que tecnologias de bancos de dados tradicionais não suportam o avanço no aumento de dados em vários ambientes, como a Web, soluções NoSQL tornaram-se cada vez mais essenciais para aplicativos corporativos. Sua proliferação nos últimos anos, trouxe um novo modelo de utilização de sistemas de armazenamento.

O modelo não-relacional forneceu maiores facilidades para a construção de aplicações Web (WYLIE, et al.; 2012). Neste capítulo serão abordados os aspectos mais importantes sobre este modelo, mas para que isso ocorra, faz-se necessário a compreensão do cenário que o antecede, na qual era dominado pelas tecnologias tradicionais de manipulação e armazenamento de dados relacionais, e só então, realizar a análise detalhada sobre os conceitos, métricas e ferramentas que incorporam este ambiente de tecnologias “Not only SQL” (NoSQL).

2.1 Modelo NoSQL

No mundo que compreende as tecnologias de banco de dados, a primeira noção que precisa-se inferir é o fato de que os dois modelos, tradicional (relacional) e não tradicional (não-relacional) possuem características especificas que os promovem, porém no mundo corporativo, não há a melhor tecnologia de banco de dados, e sim, a que melhor atende as necessidades de clientes, funcionários, empresas ou organizações. O intuito deste capítulo não é compará-los sobre o ponto de vista de performance, usabilidade, importância ou nível de complexidade, e sim exemplificar de maneira simples, algumas características que diferem os dois modelos.

Desde sua concepção em 1970, o modelo relacional de dados (ou modelo tradicional), tem sido utilizado em larga escala pelos sistemas de gerenciamento de banco de dados, tais como SQL Server, Oracle, MySQL e diversos outros, seu surgimento deu-se como forma de substituir os modelos hierárquicos e de redes, permitindo que fossem criadas relações (ou tabelas), com linhas (tuplas) e colunas (ou atributos), de forma a "acomodar" os registros, referente ao modelo de negócio usado (BRITO; 2010).

Na criação de modelos de negócios em banco de dados tradicionais, sua dependência está na especificação dos tipos de seus atributos, ou seja, para cada atributo criado, é necessário especificar o tipo de valor que será aceito. Outra característica presente nesses bancos de dados é a criação de chaves que permitem a identificação de suas tuplas especificas, além de possibilitarem a criação de relacionamentos com outras tabelas.

Nas arquiteturas SQL, as transações, são uma das formas de garantir a integridade dos dados, proporcionando consistência dos mesmos. Essa característica é também conhecida como ACID (Atomicidade, Consistência, Isolamento e Durabilidade), porém seu dimensionamento demostra alguns empecilhos em relação aos diferentes aspectos da alta disponibilidade em sistemas distribuídos (MONIRUZZAMAN, HOSSAINS; 2013).

Com o aumento de uma demanda generalizada por soluções, e a relativa facilidade de desenvolvimento de novos sistemas, levou ao florescimento de novos modelos bancos de dados. Em contrapartida, aos modelos relacionais (baseados em interfaces SQL), os modelos NoSQL possuem uma abordagem diferente em relação a manipulação de dados, pois a associação é feita por intermédio do modelo de chave-valor, assim como um Objeto JavaScript, ou um Dicionário em Python, ou ainda um Hash em Ruby, ou seja, não há a necessidade de informar ou especificar por meio de um esquema (modelo de dados) pré-estabelecido, pois cada registro pode conter um conjunto diferente valores nomeados (WARDEN, pg. 13, 2011).

Como especificado, pode-se entender o modo de como os dados são dispostos na estrutura NoSQL com as especificações de chaves-valores. Os identificadores alfanuméricos são entendidos como "chaves" e seus valores associados em tabelas simples são entendidos como autônomos (ou hashs), podendo ser sequências de caracteres de texto simples ou listas mais complexas ou conjuntos, o que pode ser observado na Tabela 1. Esta simplicidade em organizar as informações, permite que, grandes quantidades de dados distintos possam ser armazenadas rapidamente, para a implementação de tarefas como o gerenciamento de perfis de usuários.

Tabela 1 - Representação de Chave-Valor banco NoSQL - Atributos de usuários

Chave	Valor (Atributos)
1	Nome: João Silva Senha: 1234 Idade: 20 Profissão: Programador Salario: 2.000
2	Nome: Maria Soares Senha: 5678 Idade: 27 Profissão: Analista de Projetos Salario: 5.000
3	Nome: José Padilha Senha: 6578 Idade: 24 Profissão: Design Salario: 2.000

Algumas organizações utilizam este modelo de chave-valor em suas aplicações, por permitirem o acesso rápido as informações. Um bom exemplo disso é a *Amazon* (Amazon.com é uma empresa multinacional de comércio electrónico), que utiliza seu próprio sistema de banco de dados não-relacional, chamado de Dynamo, um sistema altamente escalável para o armazenamento de dados de alguns dos principais serviços da empresa. Outras empresas utilizam o modelo também, como o *LinkedIn* que utiliza o sistema *Voldmort* que segue o mesmo padrão de chave-valor (MONIRUZZAMAN, HOSSAINS; 2013).

Uma forma de organizar e manipular os dados em bancos NoSQL, é pela orientação por colunas (LEAVITT; 2010), ao invés de armazenar conjuntos de informações em uma tabela fortemente estruturado de colunas e linhas com campos uniformizados para cada registro, como é o caso com bancos de dados relacionais, bancos de dados orientados por colunas contêm uma coluna extensível de dados relacionados. Seguindo essa característica o Facebook desenvolveu o Cassandra, que possuía uma alta performance para alimentar suas aplicações, por este tipo de orientação, já a Fundação de software da Apache criou também o Hbase, a partir de uma distribuição *open source* do Google's Big Table.

Ainda na compreensão do modelo não-relacional, existe outro aspecto que não foi abordado, a orientação por documentos, que ao em vez de tabelas com estruturadas com campos uniformizados para cada registro, é possível adicionar quaisquer números de campos de qualquer comprimento em um documento. A fundação Apache Software hospeda CounchDB como uma fonte aberta de banco de dados escalável escrito *Erlang* – é uma linguagem de programação, desenvolvida inicialmente para suportar aplicações distribuídas – e acessível a partir de qualquer browser. Outro banco de dados é o MongoDB, um banco de dados de documentos, e de código aberto alta escalabilidade e facilidade em seu uso.

A maioria dos bancos de dados NoSQL são de código aberto, o que reflete na evolução do mercado global de software, tendo em vista o que todas essas organizações tecnológicas estão desenvolvendo e disponibilizando no mercado. Em sua compreensão, a opção realizada pelas organizações nos requisitos de escolha dos modelos não-relacionais em detrimento de um SGBD tradicionais, são as suas características, tais como (1) Escalonamento; (2) Consistência; (3) Disponibilidade; descrito a seguir:

1. Capacidade de escalonamento, pois o modelo apresenta flexibilidades em relação a inclusão transparente de outros elementos, por não possuir um esquema definido, assim à autonomia na manipulação dos dados;
2. Ao realizar a escolha pelo modelo “Not only SQL”, a arquitetura começa a sofrer perda a consistência, devido a sua alta flexibilidade. Porem segundo o Teorema de Eric Brewer, conhecido como CAP (Consistency, Availability e Parttion Tolerance ou Consistência, Disponibilidade e tolerância a particionamento), explica que em sistemas distribuídos somente duas, destas três vertentes, serão aderidas simultaneamente; e
3. Significa o alto grau de distribuição de dados, que é realizado para suportar um grande número de solicitações por parte de um sistema, evitando que o mesmo fique menos tempo indisponível.

Com a evolução de técnicas de desenvolvimento que aceleraram a procura por soluções que garantem essas características, implementadas pelas grandes organizações do mercado tecnológico e com o surgimento de tecnologias de Big Data, impulsionaram o crescimento do modelo NoSQL, permitindo sua evolução e de suas tecnologias.

2.2 Tecnologias de NoSQL

As mudanças ocorridas neste novo cenário levaram ao desenvolvimento de soluções

que proporcionassem as ferramentas necessárias para a compreensão deste modelo. Sendo assim, diversas empresas aderiram ao paradigma dos bancos de dados não relacionais, fortalecendo cada vez mais suas características tão singulares, e modificando as estratégias de armazenamento.

Com a difusão desta tecnologia ocorreu o surgimento de inúmeros sistemas de gerenciamento de bancos de dados não-relacionais, alguns já citados neste capítulo, oferecendo, segundo Warden (2013), as ferramentas essenciais para a manipulação de dados, diante da nova interface, ferramentas como (1) MongoDB, (2) CouchDB, (3) Cassandra, (4) Redis, (5) BigTable, (6) HBase, (7) Hypertable, (8) Voldmort, (9) Riak e (10) ZooKeeper, são descritos na sequência:

1. Especialmente desenvolvido para trabalhar-se com grandes conjuntos de dados, este banco de dados NoSQL, oferecer algumas vantagens para os desenvolvedores, ao apresentar, baixos esforços de manutenção. Utiliza a orientação a documentos, com registros que se parecem com objetos JSON, com a capacidade de armazenar e consultar em atributos aninhados, ou seja, suas consultas são encapsuladas em seus métodos;
2. Um projeto de código aberto da Apache Foundation, o CouchDB, é um banco de dados orientado a documentos JSON, altamente concorrente e projetado para ser facilmente replicável horizontalmente em inúmeros dispositivos, além de permitir o armazenamento de documentos JSON por sua interface RESTful;
3. Iniciado como um projeto interno do Facebook, logo, o sistema Cassandra também tornou-se open-source. É um banco de dados de tempo de execução, ideal para aplicações cujos os domínios sejam a Web, devido a sua alta escalabilidade. Alguns de seus recursos destacam-se em relação a outras bases de dados, como esquema flexível (possibilitando que o usuário acrescente ou remova mais elementos conforme for a necessidade de sua aplicação), escalabilidade (é a habilidade de escalar suas aplicações horizontalmente, assim não é necessário iniciar os processos, alterar consultas ou manipular manualmente os dados) e consciência de múltiplos datacenters (com isso, é realizar backups da base de dados, evitando que em caso de desastre não perca-se os dados);
4. Duas características fazem com que o Redis destacar-se em relação aos outros sistemas, que são, ele armazena os dados na memória *Random Access*

Memory (ou memória RAM) e possui a capacidade de trabalhar com estruturas de dados bem mais complexas, devido ao mesmo ser referenciado a servidores, podendo trabalhar com *strings*, *hashes*, conjuntos organizados de dados, entre outros. O Redis é escrito em ANSI C – linguagem C de programação - e funciona na maioria dos sistemas Portable Operating System Interface (POSIX) como Linux, sem dependências externas;

5. Com uma estrutura mais complexa e uma interface de muitos armazenamentos de dados NoSQL, o BigTable, possui uma hierarquia de acesso multidimensional. Sua estrutura é semelhante aos bancos de dados relacionais tradicionais, em que cada tabela é dividida em várias linhas, com cada linha endereçada com uma sequência de chave única. Os valores são posicionados dentro dessas linhas e organizados em células, com cada célula sendo identificada em uma coluna, que possui um *timestamp* como nome. Este banco de dados é utilizado pela Google para o gerenciamento de *petabytes* de informações;
6. O sistema Hbase foi desenvolvido baseado no BigTable, assim ele suporta a mesma estrutura de dados de tabelas, chaves, famílias de colunas, nomes de colunas, *timestamps*, e os valores das células;
7. Hypertable é outro clone de código aberto do mesmo BigTable. Ele é escrito em C ++ (linguage de programação baseada em C), ao invés de Java como HBase, porém possui uma alta performance;
8. Desenvolvido pela organização LinkedIn e baseado no Dynamo (banco de dados desenvolvido pela Amazon e utilizado no controle de informações de seu sistema de *e-commerce*, mais especifico em seu "carrinho de compras"), utilizando-se de estruturas de hashes, concede pesquisas rápidas, e tem controle para manipular valores inconsistentes;
9. Riak é *open-source,* e possui algumas características presentes em quase todos os bancos de dados citados, como baixa latência, disponibilidade, tolerância a falhas, simplicidade em suas operações e escalabilidade; e
10. O ZooKeeper foi originalmente construído pela organização Yahoo! para tornar simples que as aplicações da empresa acessassem as configurações de forma robusta e de fácil compreensão, mas desde então tem crescido ofertando uma série de características que ajudam a coordenar o trabalho dentro de aglomerados distribuídos de dados. Criado para funcionar de forma

distribuída por meio de um número de máquinas, e projetado para oferecer uma rápida leitura em detrimento das gravações de dados captados;

Todas essas tecnologias citadas refletem uma importante indagação referente ao imensurável aumento na fabricação de dados em diversos domínios, e tornam-se cada vez mais essenciais no cenário atual, ao qual diz respeito sobre as tecnologias de BigData, por proporcionarem a capacidade de armazenamento e manipulação de grandes conjuntos de dados.

2.3 NoSQL no cenário de Big Data

No contexto atual, bancos de dados não relacionais tornaram-se essenciais para a manipulação de grandes volumes de dados de diversos domínios, pois como apresentado no capítulo 1, as ferramentas de Big Data não trabalham com apenas dados estruturados, mas estende-se aos semiestruturados e não estruturados, possibilitando uma análise de diversas fontes heterogêneas, como a Web, rede de sensores e etc.

A manipulação de dados distintos e heterogêneos no contexto de Big Data é utilizado bancos de dados NoSQL, que proporcionam algumas facilidades, tais como, o trabalho com estruturas não-relacionais, distribuídas, escaláveis horizontalmente, altas capacidades de trabalharem com grandes volumes de dados, velocidade e variedade sem à necessidade de esquemas estruturados.

Organizações que coletam grandes quantidades de dados não-estruturados, de acordo com Moniruzzaman e Hossains (2013), estão cada vez mais utilizando recursos não-relacionais, devido aos mesmos terem o foco no processamento analítico de conjuntos de dados em larga escala, oferecendo uma maior escalabilidade sobre hardware das maquinas utilizadas para o armazenamento.

O desenvolvimento dos bancos de dados NoSQL como o BigTable, HBase e o Cassandra, permitiram a elevação do desenvolvimento de estruturas escaláveis horizontalmente e armazenamentos de dados não-relacionais distribuídos, desfavorecendo os recursos computacionais que utilizavam bancos de dados centralizados relacionais.

Os bancos de dados orientados por documentos são ótimos recursos para trabalharem com o armazenamento e gerenciamento de grandes coleções de dados, alimentando as tecnologias e ferramentas de Big Data (OREND; 2010), por eles trabalharem com documentos de textos, documentos XML, entre outros. Muito utilizados em organizações que utilizam arquiteturas de consultas, como sistemas baseados em Service-Oriented Architeture

(SOA). Alguns sistemas de gerenciamento de dados não-relacionais, como já descritos anteriormente, fornecessem esta possibilidade de manipulação de documentos como o MongoDB e o CouchDB.

Uma outra abordagem que bancos de dados como o MongoDB e o CouchDB suportam é o MapReduce, é um modelo de programação para o processamento e a geração de grandes volumes de dados. Este modelo é dividido em duas funções, *map* que processa um par de chave-valor, para gerar um conjunto intermediário de chave-valores pares, e *reduce*, uma função que combina todos os valores intermédios associados com a mesma chave intermediária (DEAN, GHEMAWAT; 2004).

Segundo Dean e Ghemawat (2004), o modelo oferece o paralelismo para os programas escritos de acordo com suas funcionalidades, devido aos mesmos serem geralmente executados em *clusters* (conjunto de computadores, que utiliza um sistema operacional especial classificado como distribuído). Na função *map* as invocações (ou chamadas) são distribuídas entre várias maquinas dividindo automaticamente os dados de entrada os quais serão processados, e as separações das entradas podem ser executadas em paralelo em diferentes maquinas, e por fim, na função *reduce* as invocações são distribuídas dividindo os espaços intermediários por outra função que os particiona.

Uma variante do *Map and Reduce* é chamado de “Incrementais MapReduce”, que fornece capacidade de trabalhar com situações críticas de *streaming* (ou fluxo) de armazenamento de dados. O “Incremental MapReduce”, significa que o mesmo pode ser incrementado apenas quando os novos dados entram, assim os seus resultados são validados para todo o armazenamento de dados. CouchDB suporta esta variante nativamente e MongoDB apoia-o indiretamente com alguma contabilidade adicional.

As tecnologias de BigData estão cada vez mais em evidência, e com o seu crescimento surgem ferramentas e tecnologias que as apoiam, o NoSQL não é uma exceção, surgindo como um novo paradigma a fim de explorar as condições impostas sobre este novo termo, como a análise de dados estruturados, semiestruturados e não estruturados, logo tornou-se uma ferramenta indispensável para a manipulação de grandes volumes de dados.

2.4 Considerações Finais do Capítulo

Com uso das tecnologias de NoSQL levaram ao aprimoramento dos conceitos de utilização e reutilização de dados de diversos cenários, desfavorecendo as tecnologias baseadas em interfaces SQL tradicionais, em relação a alguns pontos chaves, como a ausência

de restrições convencionais de armazenamento, pois um dos fatores mais importantes do seu surgimento é a persistência poliglota que permite a utilização de mais de uma base de dados em uma mesma aplicação, o que facilita na sua escalabilidade, o tempo de resposta, assim os bancos de dados deixam de ser o gargalo e se transforma na principal solução, elevando o nível da aplicação independente do modelo de negócio (OREND; 2010).

O que pode-se inferir sobre essa tecnologia é a sua capacidade de armazenamento de dados distintos sem a utilização de esquemas relacionais, assim como observado em ambientes tradicionais de bancos de dados transformando diversos cenários e modelos de negócios.

A utilização de bases de dados NoSQL para a manipulação de grandes volumes de dados é uma alternativa para os cenários de Big Data, devido ao mesmo trabalhar com dados de diferentes formatos, e sua utilização está incorporado a este quesito. Com a integração desta tecnologia na plataforma desenvolvida neste trabalho podendo vir a garantir uma alternativa a manipulação de grandes volumes de dados.

3 EXTRAÇÃO DE DADOS

Os avanços das tecnologias Word Wide Web (WWW) possibilitaram a geração de inúmeras fontes de dados neste cenário. A Internet possui muitos dados de relevância, disponíveis em seus documentos, porém seu modelo de publicação das informações permite aos seus usuários um modo informal de publicá-las, que remete a não garantia da consistência desses dados, dificultando sua filtragem para subsidiar diversas áreas de conhecimentos.

Conforme D'Andréia (2006) destaca, tamanha a diversidade de informação que possibilita a potencial multiplicidade de vozes manifestando-se sobre um tema, quanto faz com que seja ainda mais importante lançar um olhar criterioso sobre o universo Web. A facilidade de publicação e acesso a informações do seu modelo gerou o círculo virtuoso que a retroalimenta, promovendo seu crescimento. No entanto, este cenário trouxe consigo dificuldades para o uso efetivo das informações (MENDONÇA, 2003).

Mendonça (2003) afirma ainda que ferramentas de busca - como o Google - tem como principal objetivo compensar este modelo de publicação, a fim de permitir uma alternativa para a recuperação das informações espalhadas na internet, oferecendo aos usuários uma interface, que por meio desta, é possível informar palavras-chaves referentes a pesquisa, como por exemplo notícias, lugares, entre diversas outras expressões chaves e com base nestes dados são julgados as altas probabilidades dos mesmos estarem presentes em documentos relevantes da pesquisa em questão.

Porém as buscas realizadas por ferramentas de busca evidentemente não são realizadas exatamente na Web e sim por catálogos internos de informação, que são obtidas previamente por intermédio de mecanismos de localização que efetuam a indexação das páginas Web e ranqueias pela sua relevância, de forma a auxiliar e garantir a melhor qualidade nas buscas. Informações relevantes na Web estão localizadas em documentos HTML e despertam o interesse da comunidade cientifica e de organizações.

Com isso algumas áreas sugiram para auxiliar esta demanda crescente como é o caso da extração de dados, que neste capítulo, será abordado seu conceito, objetivos e desafios de modo a garantir seu entendimento gerando um olhar mais amplo sobre a mesma, além de revelar algumas das técnicas que são utilizadas, bem como a que será apresentada neste trabalho.

3.1 Conceitos de Extração de Dados

HTML e XML são linguagens de marcação para documentos Web, pelo uso de *tags* (elementos de marcação de texto). Essas tecnologias são utilizadas para a representação de dados dos mais variados formatos. O XML não possui *tags* estáticas, ou seja, chaves referentes ao atributo, elas são especificadas por cada autor, assim é necessário que outros que pretendem consumir suas informações tenham a capacidade de compreender sua intenção, em contrapartida, as *tags* presentes no HTML são especificadas pela autoridade que a criou e seu órgão especializado, a W3C.

Com isso muito desenvolveu-se por meio destas tecnologias, que possibilitaram a inserção de materiais antes encontrados somente em bibliotecas, para o ambiente Web, entre estes valem apena destacar, jornais, revistas, documentos impressos como livros históricos e artigos científicos, entre inúmeras fontes de conhecimento, que proporcionaram um aumento exacerbado no volume de dados neste ambiente.

A falta de integração de dados em um único contexto, tornou-se complexo garantir a consistência das informações disponíveis em diversos domínios na Internet, este cenário está descrito a seguir,

> Atualmente, a informação está disponível de maneira rápida, barata e disseminada. Assim, não é de se admirar que todos se queixem do excesso de informação. Note que hoje em dia o problema não é o acesso à informação, mas sua sobrecarga. Como resultado, tem-se a necessidade de filtrar e discernir. Neste contexto, o discernimento é um fator chave, pois implica que o indivíduo deve possuir a capacidade de julgar as coisas, clara e sensatamente(...) (FILHO; DELGADO, 2003).

É importante salientar que o problema abordado neste texto não é a imensa gama de informação disponível na Web, e sim a sua inserção em contextos, sendo capaz de recuperá-las de maneira categórica e precisa. O objetivo principal da extração de dados a partir da WWW é fornecer uma visão integrada de dados autônomos de fontes de informação, relatado por Wolfgang e George (2000).

Segundo Mendonça (2003), fontes de informação são sistemas que respondem a consultas, retornando uma resposta apropriada para cada consulta submetida, no âmbito da Internet estas fontes possuem formatos textuais, que são manipuladas pelo paradigma *request-response* implementado pelo protocolo Hypertext Transfer Protocol (HTTP), geralmente utilizam os formatos semiestruturados apresentados no início deste capítulo como HTML e XML.

Com a utilização da extração de dados para a localizar, coletar e organizar dados de interesse apresentados nos formatos citados, é possível o enriquecimento de grandes bases de dados, permitindo a realização de consultas e cruzamento dos mesmos, que não eram possíveis devido as interfaces de consulta pré-estabelecidas nas fontes de informação, possibilitando a maior integridade das mesmas já disponíveis, facilitando a construção de agentes inteligentes.

3.2 Objetivos da Extração de Dados

Extração de dados tem como objetivo fornecer maneiras de obter os dados de diferentes fontes heterogêneas, como de documentos HTML, de modo a oferecer informações para um banco de dados ou algum outro aplicativo.

Esta tarefa é realizada por um sistema de extração de dados que podem ser divididos em cinco diferentes funções, definidos por Baugartner, Gatterbauer e Gottlob (2009): (1) Interação com a Web; (2) Apoio à geração e execução; (3) A programação; (4) Transformação de dados; (5) Analise dos resultados obtidos,

1. Compreende principalmente a navegação geralmente pré-determinada a páginas Web, que contenham as informações desejadas;
2. O programa que identifica os dados desejados em páginas estabelecidas, extrai os dados e transforma-os em um formato estruturado;
3. Permite que a aplicação possua o desenvolvimento voltado às suas respectivas páginas de destino;
4. Inclui a filtragem, transformação, refino e integração de dados extraídos de uma ou mais fontes e estruturar o resultado de acordo com um formato de saída desejado (geralmente XML e JSON); e
5. Retornar o resultado de dados estruturados para aplicações externas, tais como sistemas de gerenciamento de banco de dados, Data Warehouses, sistemas de software de negócios, sistemas de gerenciamento de conteúdo, sistemas de apoio à decisão, editores de Really Simple Syndication (RSS), servidores de e-mail, Short Message Service (SMS) ou servidores. Em alternativa, a saída pode ser usada para gerar novos serviços Web de fontes já existentes e em contínua mudança.

Os sistemas de extração de dados na Web são amplas classes de aplicações de software que visam a extração de informações a partir da mesma (Laender et al., 2002).

Geralmente interagem com um recurso Web e extraem os dados guardados no mesmo, por exemplo, se um recurso é uma página HTML, a informação extraída pode consistir tanto em elementos da página como em seu texto por completo. Eventualmente, os dados extraídos podem ser processados, convertidos no melhor formato estruturado conveniente e armazenado para um uso posterior (FERRARA et al., 2012).

3.3 Desafios da Extração de Dados

Como já mencionado neste capítulo o processo de publicação das informações influenciou diretamente na expansão da Web, forneceu a área de extração de dados fontes heterogêneas de dados para a captura e armazenamento dos mesmos, porém, muitos desses meios oferecem dificuldades e desafios para a execução dessa tarefa tanto pelas técnicas de extração quanto pelo desenvolvimento de um projeto de extração.

Segundo Silva, Barros e Prudêncio (2005) documentos Web podem ser classificados como estruturados (que apresenta um formato rígido, e as informações do documento são geradas a partir do banco de dados, além de facilitar sua extração utilizando regras baseadas em delimitadores e/ou ocorrência de termos), semiestruturados (são documentos estruturados, mas podem apresentar irregularidades como campos ausentes ou com valor nulo, variações na ordem dos dados, e ausências de delimitadores) e não-estruturados (ou livres, este tipo contêm basicamente sentenças em alguma linguagem natural, o que inviabiliza a extração com base somente na formatação).

A análise realizada sobre o texto presente nos documentos Web em que pretende-se extrair os dados, é um dos fatores cruciais para o desenvolvimento posterior. Mendonça (2003) aborda alguns fatores importantes quando almeja-se a extração em arquivos semiestruturados, em relação a alguns atributos apresentados nos mesmos, os quais são abordados os atributos ausentes (que podem ser representados por conjunto de dados nulos ou ausentes, ou seja, faltando até mesmo seus marcadores e delimitadores), atributos multivalorados (assim como o que é apresentado nos modelos relacionais em banco de dados, esta maneira de apresentar os dados remete a sua não normalização devido à falta de padrão), diferentes ordenações dos atributos (em um documento HTML as informações não são apresentadas uniformemente, isto é, os dados da página seguem o padrão definido pelo autor) e delimitadores disjuntivos (compreende os documentos HTML cujo os seus delimitadores não fazem referência aos valores que estão representando, por exemplo, uma *tag "span"* poderá conter textos, números, entre outros, o mesmo não ocorre em documentos XML).

Em um projeto de sistema de extração de dados, muitos fatores devem ser levados em conta, de acordo com Ferrara et al. (2012) alguns desses fatores são independentes do domínio de aplicação específica em que pretende-se realizar a extração de dados da Web e outros, dependem fortemente das características específicas do domínio da aplicação, como consequência algumas soluções tecnológicas que parecem ser eficientes em alguns contextos não são adequadas em outros.

Outro ponto importante abordado pelos mesmos autores referente aos desafios do projeto de sistemas de extração de dados na Web, em que é apresentada algumas especificações importantes em sua concepção, corresponde a:

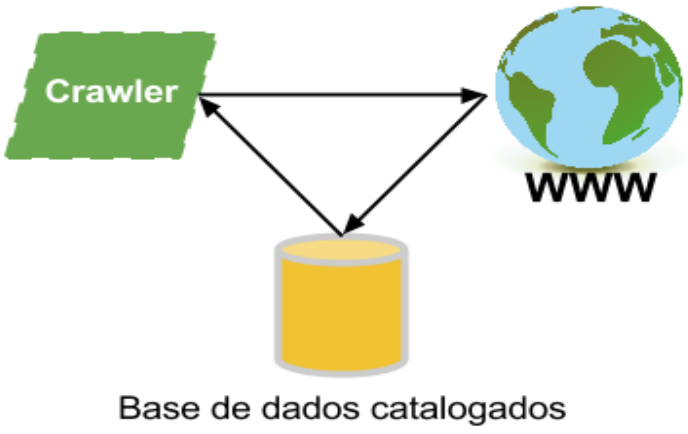
- Especialistas em técnicas de extração de dados, pois faz se necessário o estabelecimento sobre o processo de construção de procedimentos altamente automatizados de extração de dados e da exigência de alcançar alta precisão e performance;
- Técnicas de extração de dados da Web, devem possuir a capacidade de processar grandes volumes de dados em um curto espaço de tempo, esta capacidade reflete nas necessidades emergentes da Inteligência de Competitiva;
- As aplicações no campo da Web social, ou seja, que lidam com dados relacionados a pessoas, devem garantir a privacidade, qualquer tentativa de extração desses dados (mesmo que não intencionadas) devem ser anunciadas e identificadas;
- Algumas abordagens como aprendizagem de máquina, que corresponde na marcação manual das páginas Web, muitas vezes exige um grande conjunto de treinamento para reconhecimento de padrões referentes ao conjunto de domínios, em geral, a tarefa de rotulagem de páginas, demanda de tempo, auto investimento e sujeita a erros e por consequência, em muitos casos, não se pode assumir a existência de páginas marcadas; e
- Algumas fontes de dados, em que as informações são extraídas podem sofrer alterações estruturais significativas sem aviso prévio que acarretará no mau funcionamento da aplicação, é necessário que o mesmo tenha a flexibilidade em detectar essas mudanças.

3.4 Técnicas de Extração de Dados

Obter informação a partir da Web pode abranger a compreensão de outros sistemas de extração de dados, entrando em questão as ferramentas de busca, que a partir de uma consulta do usuário efetuam a localização de documentos ordenados pela relevância da pesquisa, este é outro contexto distinto que precisa ser diferenciado, de modo a evitar assim confusões entre os termos de extração de dados.

Algumas das ferramentas de busca utilizam como forma de extrair informações programas capazes de percorrer a Web automaticamente catalogando e classificando *URLs*, que com as informações obtidas, serviram para o armazenamento em repositórios locais, esses programas são chamados de Web Crawlers representado na Figura 2. Porém este tipo de software utiliza uma navegação entre as páginas do tipo "força bruta", isto é, não há especificações (filtros) que identifiquem os links que o software deverá acessar, pois ele move-se automaticamente de site para site, seguindo os links embutidos em seus documentos copiando tudo que lhe for possível.

Figura 2 - Representação do funcionamento de um Crawler.

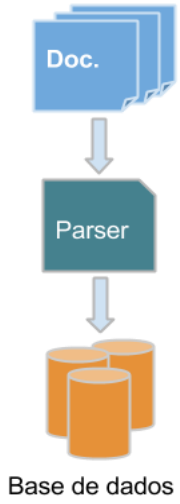


Fonte: Próprio Autor

Um importante fato ocorrido em junho de 2013 que atingiu rapidamente repercussão global, foi o vazamento de documentos secretos referentes ao governo dos Estados Unidos da América, e seu principal acusado Eduard Snowden, que haveria segundo especialistas da comunidade de inteligência utilizado os recursos de um Web Crawler para capturar tais documentos (Bamford, James; 2014) (Sanger, E. David; Schmitt, Eric; 2014).

Outro software capaz de realizar a extração de dados em documentos são os Parsers, um programa que espera receber dados de uma maneira estruturada, de forma que informações consigam ser extraídas mecanicamente, este tipo de abordagem oferece pouca inteligência, pois não são capazes de realizar a extração em ambientes não estruturados. Essa característica desta técnica pode ser observada na Figura 3.

Figura 3 - Representação da funcionalidade de um Parser.



Fonte: Próprio Autor.

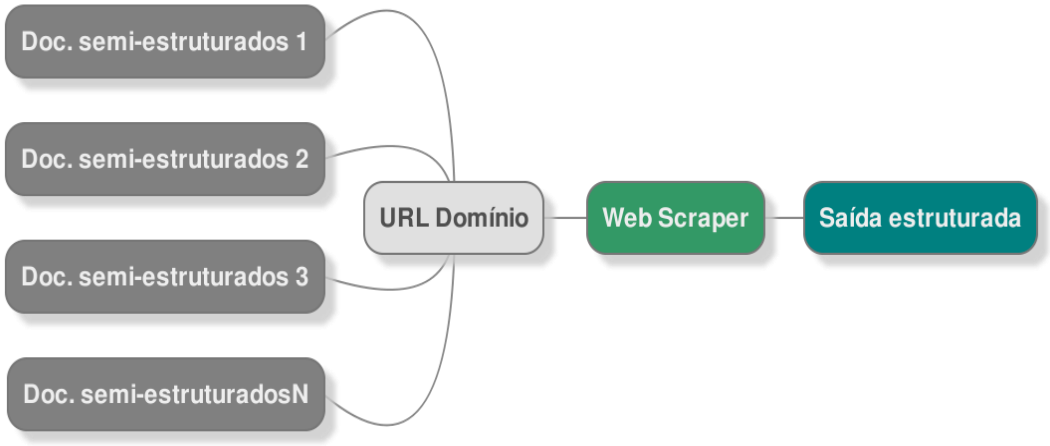
Sistemas de extração de dados visam localizar e extrair, de forma automática, informações relevantes em um documento ou coleção de documentos, contendo textos em línguagem natural, e estruturar tais informações para os padrões de saída, a fim de facilitar sua manipulação e análise (GRISHMAN, 1997). Neste contexto é ressaltado o programa caracteriza esta ideia, o Data *Scraping* é também um software capaz de extrair dados da saída de um outro programa, esse modelo é mais conhecido popularmente nos dias atuais como Web Scraping, um software capaz de extrair dados de documentos Web, com base em estruturas HTML ou XML.

Web Scraping (conhecido também como Screen Scraping) é uma técnica de captura de dados a partir de *sites*. Está intimamente relacionado com a indexação da Web, que indexa mais detalhes sobre a mesma utilizando scripts ou *bots* (robôs de busca) e é uma técnica mundial adotado pela maioria dos motores de busca. Em contraste, concentram-se mais na transformação de dados não estruturados na internet, geralmente em formato HTML, em dados estruturados que podem ser armazenados e analisados em um banco de dados central, local ou planilha. A extração de dados utilizando este software é realizada pela simulação da

navegação de um ser humano, que podem incluir a comparação on-line de preços, monitoramento de dados, detecção de mudanças em Web sites além de pesquisas e integração de dados (MASNICK; 2009).

A Figura 4 ilustra um exemplo direto e simplificado da ação de um Web Scraping, com base na saída de documentos semiestruturados, neste caso páginas HTML. São realizadas as análises destes documentos de um único domínio possuindo dados de interesse que servirão para constituir uma saída estruturada das informações extraídas das páginas. Por meio do download de páginas, documentos, imagens ou qualquer tipo de arquivo que com simples requisições HTTP, as mais conhecidas são POST e GET.

Figura 4 - Representação do funcionamento de um Web Scraping.



Fonte: Próprio Autor

A principal atividade que um Scraping deve desempenhar é a navegação Web que visa simular as ações exercidas por um *Browser* (Google Chrome, Mozilla Firefox, Internet Explorer, etc.). A navegação consiste em "abrir" diferentes páginas de um site com o objetivo de coletar dados ou executar consultas. Com isto, é realizado "*downloads*" das páginas, com simples requisições HTTP, utilizando os métodos POST e GET. O método GET é usado para recuperar qualquer informação referenciada por uma URL, permitindo o primeiro reconhecimento da mesma. Mas o POST vai além, e é usado para enviar um conjunto de dados para um serviço especificado por uma URL e recuperar as informações resultantes do processamento desses dados, o que pode ser compreendido como o envio de dados para uma consulta em um formulário.

Além desses dois métodos, outro que pode-se destacar para o desenvolvimento de um Scraping é o HEAD, é idêntico ao GET, mas o servidor não retorna o conteúdo no corpo da página referenciada pela URL requisitada, este método é utilizado principalmente para

recuperar meta-informações presentes no *header* da resposta na requisição, e pode conter informações como *Host, User-Agent, Accept-language* entre diversas outras, podendo ser utilizada pelo programa no envio de parâmetros ocultos nos domínios.

3.5 Considerações Finais do Capítulo

A extração e recuperação de dados a partir da Internet, possibilita a coleta dos dados presentes neste ambiente, permitindo que os mesmos sirvam para alimentarem grandes bases de dados, a fim de serem utilizadas posteriormente por sistemas que apoiam a tomada de decisão. A utilização de soluções como Web Scrapings podem garantir estas características, neste trabalho optou-se pela escolha desta técnica, e por meio desta, obteve-se alguns resultados e contribuições relevante para a área de extração de dados, o que pode agregar a soluções já estabelecidas na área.

4 PLATAFORMA DE EXTRAÇÃO E RECUPERAÇÃO NA WEB NO CONTEXTO DE BIG DATA

A grande massa de dados disponibilizados em diversos domínios, como a Web, permite aos especialistas da área de dados explorarem as informações disponíveis nela, com o uso das técnicas de extração, armazenamento e análise das informações, por meio da utilização de agentes de busca, bem como os Web Scrapings. Este trabalho concentrou-se no estudo de técnicas de extração da Web, e com base neste primeiro objetivo, foi desenvolvida uma Application Programming Interface (API), que segue o estilo de arquitetura RESTful, com a principal função de extrair dados contidos nas páginas da Internet, por meio da parametrização de um agente extrator de dados auxiliando especialistas da área. Como consequência, focou-se na criação de um outro serviço, também Web, cujo seu principal objetivo era de demonstrar a utilização da API, concretizando assim em uma plataforma de extração de dados.

Nos ambientes cooperativos atuais, diversas aplicações são baseadas em arquiteturas como serviços, devido as facilidades encontradas na integração e principalmente a capacidade de escalar várias aplicações, a fim de, descentralizar o controle total da mesma, de modo a evitar as redundâncias de suas funcionalidades.

Segundo Massé (2011) os serviços Web (ou Web Services) são servidores construídos com o propósito de suportarem as necessidades de um *site* qualquer ou outro aplicativo. As APIs são serviços Web com base em interfaces de programação que permitem a comunicação de programas “clientes”, de um modo geral, estas aplicações expõem um conjunto de dados e funções para facilitar a interação entre programas de computador, permitindo-lhes a troca de informação. Pode-se entender uma API como sendo um “rosto” de um serviço Web, diretamente “ouvindo” e “respondendo” às solicitações de sistemas clientes pelos métodos contidos nos protocolos de comunicação.

O estilo arquitetônico Representational State Transfer, (REST, ou Transferência de Estado Representativo), é comumente aplicado ao *design* de APIs de serviços Web modernos. Uma API Web em conformidade a este estilo arquitetônico é descrita como REST API. A arquitetura REST abstrai como a Word Wide Web trabalha, com os métodos de comunicação e da representação de dados. Este conceito foi inicialmente descrito e implementado por Roy Fielding (principal autor do protocolo HTTP) (RICHARDSON, RUBY; 2007), assim então,

compreende-se os serviços Web que seguem as suas restrições arquitetônicas como sendo um RESTful.

O RESTful é bastante utilizado hoje por possuir os quatro verbos essenciais que o especificam, são eles GET, POST, PUT e DELETE, aos quais são frequentemente utilizadas em relação as operações de *Create*, *Read*, *Update* e *Delete* (ou CRUD, presente em qualquer aplicação que acesse as informações em um repositório ou banco de dados), e por serem utilizados para a transferência de dados em formatos semiestruturados como XML ou JSON (RICHARDSON, RUBY; 2007).

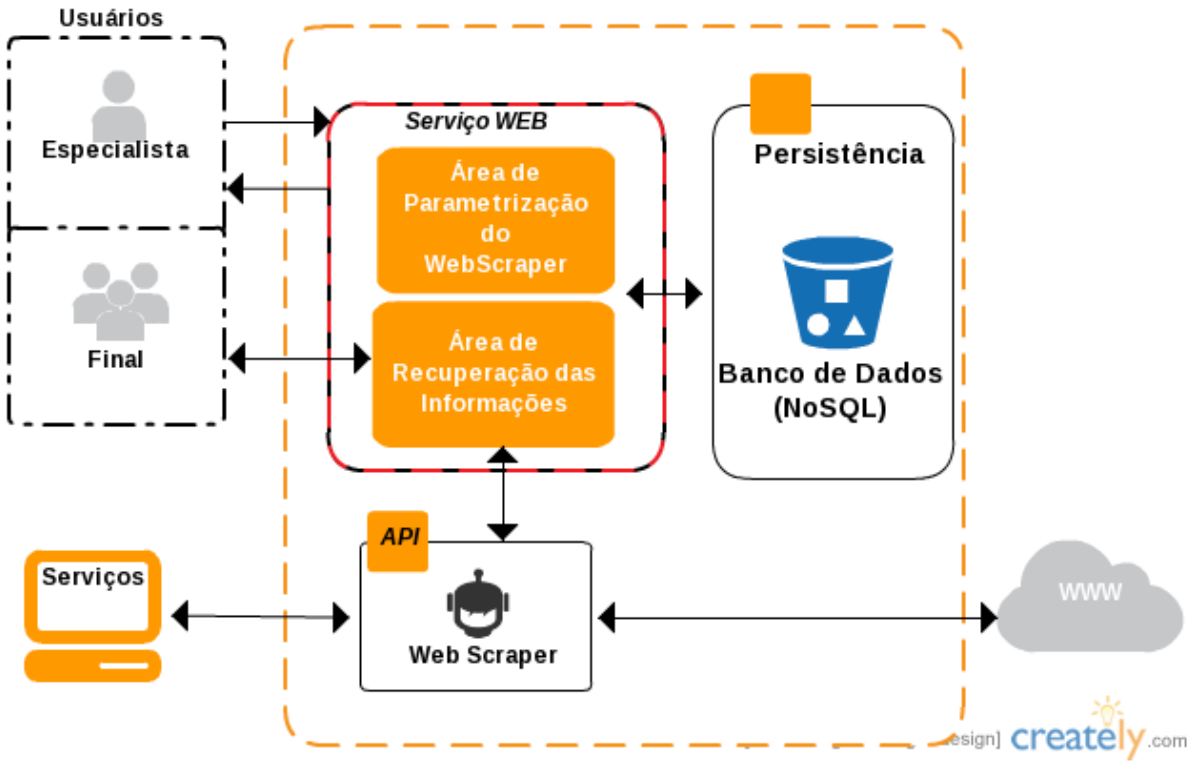
O principal produto desta pesquisa foi o desenvolvimento de uma API baseada neste modelo de representação, a qual possui recursos que garantam a confiabilidade e segurança das transferências das informações. Porém seu maior enfoque está ligado na parametrização de métodos para extração de dados, em diversos domínios da Web, a fim de extrair informações no formato de Big Data. A parametrização se dá por meio do usuário, em que o mesmo, especifica as regras de extração de seu próprio robô de busca e posteriormente irá realizar a transferência de elementos necessários para dar sequência ao processo de extração das informações.

Para o estabelecimento das técnicas utilizadas no processo de captação de dados, o usuário precisa fornecer elementos cruciais, que compreendem à, URL que pretende-se extrair os dados, elementos presentes nas páginas HTML tais como *tables, h1, divs, span*, bem como sua hierarquia ou a estrutura que encontra-se, entre qualquer outro elemento presente no corpo da página (*body*), e por fim o formato de retorno dos dados que serão extraídos, no caso o formato de envio dos parâmetros é um JSON.

No entendimento do serviço estabelecido (a API), observou-se a necessidade de desenvolver uma simples aplicação, com uma interface amigável e usual, cujo o único objetivo era de demostrar as funcionalidades e a forma que é realizada a parametrização de um agente extrator. O serviço é a exemplificação da funcionalidade de parametrização do agente, e são fornecidos os dados primários para a sua composição, e posteriormente para a sua armazenagem, em um banco de dados não-relacional. Além de possuir uma área para recuperação das informações, o qual permite que os robôs possam ser executados.

Na Figura 5 é apresentada toda a arquitetura desta plataforma, como forma de proporcionar a compreensão do mesmo. Nesta pesquisa os esforços de desenvolvimento foram direcionados para a consolidação dos métodos de extração de dados, presentes na API e concretização de seu uso por meio do serviço Web.

Figura 5 - Representação da Plataforma de extração da Web.



Fonte: Próprio Autor

A plataforma de extração de dados está dividida em duas principais aplicações, como demostrado na Figura 5, são elas (1) API e (2) Serviço Web:

1. Baseada na arquitetura *RESTful* com seu principal objetivo de extrair informações contidos em documentos HTML (sites, blogs, portais de notícias, dentre outros), mas para efetuar esta ação, a mesma possui um *Scraping*, com seus métodos parametrizados, ou seja, as regras do robô de busca, não estão inerentes a ele mesmo, e sim com usuário, fazendo com o que, a extração ocorra somente quando enviado os dados corretamente. A API fora desenvolvida na linguagem de programação Ruby e seu Framework Rails (Ruby on Rails), o que promoveu sua rápida implementação, além de possuir algumas bibliotecas especificas da própria linguagem, para a criação de robôs de busca. Como sua arquitetura está baseada como um serviço RESTful, existe a possibilidade de acesso por meio de qualquer outro serviço que especifique a URL e os seus parâmetros, com uma requisição simples; e
2. O serviço Web Cliente, que também foi desenvolvido com a linguagem Ruby on Rails, e utilizado um banco de dados NoSQL, no caso o MongoDB. Somente o serviço em questão realiza os acessos a base de dados. A aplicação

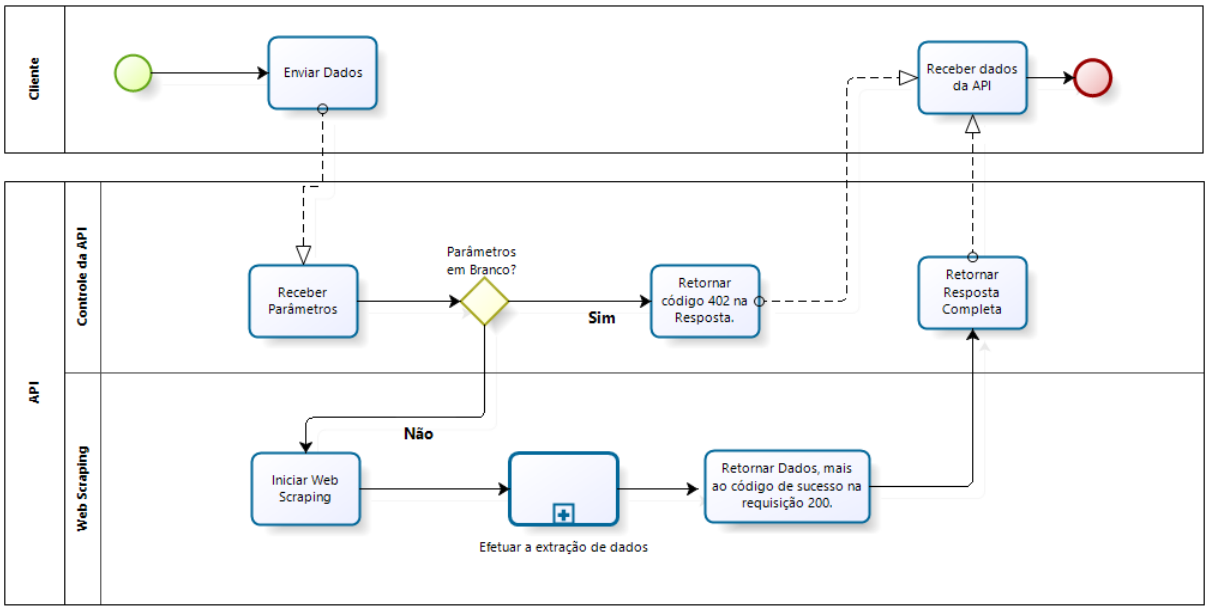
possui a única função de demostrar as funcionalidades da API, tornando-se seu cliente, por isso suas interfaces estão divididas em duas áreas principais, sendo elas, Área de parametrização do *Scraping* (responsável em manter as funções de CRUD, ou seja a manipulação dos robôs criados está contida nela, para sua utilização o usuário deve ser capaz de abstrair as informações e a maneira que é o desenvolvimento de um robô de busca) e Área de recuperação das informações (está área compreende em disponibilizar o robô criado, de modo que, outros usuários ou serviços sejam capazes de utiliza-lo).

Como exemplificado no decorrer do texto e na Figura 5, a plataforma está dívida nessas duas aplicações, a cada qual, possui suas próprias especificações. Por isso neste capítulo é importante salientar como os dois produtos deste trabalho foram desenvolvidos com a maximização de seus detalhes, fortalecendo a sua compreensão.

4.1 Especificações da API

A principal ideia para o desenvolvimento da aplicação no modelo de uma API ocorreu por meio de possibilitar a integração com outras aplicações em seu contexto. Seguindo o designer conceitual necessário para uma aplicação deste porte, o serviço contém algumas características, bem como, a especificação de cliente-servidor, demostrado na Figura 6, a qual é realizada uma requisição (*request*) por intermédio de sua URL mais o seu método, no caso um GET, neste primeiro processo são enviados os parâmetros necessários para a realização da extração de dados, os quais estão especificados na Tabela 2.

Figura 6 - Representação do modelo da Aplicação API com um Cliente qualquer.



Fonte: Próprio Autor.

Tabela 2 - Representação do recurso de acionamento da API, Utilizando o Servidor Local.

Nº	Padrão URL	Métodos	Operação	Parâmetros enviados
1	http://localhost:3000/api/v1/scraper.json	GET	Consultar	{ “url”: “URL do site”, “elements”: “span, p” }

Seguindo o estilo de arquitetura RESTful, a URL deve ser constituída seguindo alguns padrões, que estabelece qual recurso da aplicação está sendo solicitada. O ambiente do Framework Rails é propicio para o estabelecimento dessas especificações, com a configuração do arquivo chamado de “*routes.rb*” (ou rotas, se traduzir o nome do arquivo), o qual consiste nos caminhos de acesso aos recursos da aplicação, neste arquivo é especificado o método de requisição, a formatação de escrita da URL que será acessada por um usuário ou sistema, e é indicado então qual recurso será acessado, no caso deste trabalho existe apenas uma rota de acesso, apresentada na Tabela 2.

A aplicação recebe a nova requisição, verificando-se os parâmetros enviados são válidos, por meio de suas chaves, as quais correspondem a “*url*” e “*elements*”, caso um desses parâmetros não seja enviado, o programa retorna um código “402” especificado na Tabela 3. Após esta primeira verificação e com o envio dos parâmetros corretamente, é acionado o

agente extrator (*Scraping*), o qual realizará a extração, porém se em sua tentativa o mesmo falhar, é verificado os possíveis motivos de falhas, os quais podem ser, (1) Não existente; (2) Erro inesperado, respectivamente:

1. Os dados enviados para a extração, não encontram-se presente nesta página, caso isto ocorra é enviado o código 403 e os parâmetros solicitados; e
2. O site encontra-se indisponível no momento ou o mesmo bloqueou o acesso (no caso o IP da rede), devido as inúmeras requisições sequenciais, sendo assim, é retornado o código 404.

Ao final desta ação de verificação, o Cliente, ainda aguarda sua resposta, que com a execução do *Scraping*, é possível efetuar o retorno dos dados para a aplicação requisitante, para o retorno dos seguintes elementos, como informado em seguida:

- "*time*", esta chave retorna o dado referente ao tempo, ou seja a data, hora e segundo, permitindo observar o momento de acionamento da API, assim se o usuário criar alguma rotina para as requisições, o mesmo controle a data de possíveis falhas ou sucessos;
- "*params*", esta chave retorna ao usuário os parâmetros "*url*" e "*elements*", caso o mesmo não possua um controle dos dados enviados;
- "*data*", chave mais importante, pois retorna os dados extraídos no formato de texto, para serem inseridos nos repositórios ou bases de dados utilizados, caso ocorra algum problema na extração é retornado o código de falha, como apresentado na Tabela 3, para cada possível problema; e
- "*menssage*", retorna o código 200 referente a requisição e ao sucesso da consulta, caso por algum motivo o serviço esteja inoperante é informado o código 400.

Tabela 3 - Representação dos códigos presentes na requisição da API

Código	Mensagem	Descrição do Retorno
200	*Success action.*	A requisição obteve sucesso esperado, o qual significa que a extração ocorreu com êxito.
402	*Invalid Parameters.*	A requisição retornou o código pois foi enviado os parâmetros incorretos.
403	*The data does not exist in the source site's database.*	A requisição retornou uma falha pois não foram encontrados os elementos solicitados no site.

404	*Error Unexpected.*	A requisição retornou um erro inesperado o que pode ocorrer caso o site esteja inoperante, ou devido a muitos acessos sequencias API tenha sofrido bloqueio do mesmo.
400	*Internal Error.*	A requisição retornou um erro interno da aplicação, o que pode significar que a sua codificação esteja com problemas.

Contudo para que o retorno seja realizado conforme exemplificado acima, é necessário o acionamento do principal produto deste trabalho, o Web Scraping, que realiza a extração dos dados composta por algumas atividades essências.

4.1.1 Atividade do Web Scraping

A principal atividade que um Web Scraping – que a partir de agora será abordado como *Scraping* – deve desempenhar é a navegação Web, relatado no terceiro capítulo. O modelo deste software desenvolvido para o trabalho, a navegação Web é realizada com base no método GET, que realiza o "*download*" da página, permitindo a recuperação de qualquer informação da URL, e seu reconhecimento. Em diversas linguagens de programação de alto nível, a biblioteca que proporciona a utilização deste e outros métodos é a "Net::HTTP*"*, presente no Ruby, e responsável por manter os métodos do protocolo HTTP. Com ela, um fluxo de execução do *Scraping* poderia ser, acessar uma URL, coletar links relevantes na mesma, acessar estes links e extrair os dados das páginas.

Para a navegação Web uma importante capacidade é a armazenagem de *cookies*, que são pequenos arquivos de textos que efetuam a troca de dados entre o navegador "Cliente" e o Servidor, neste arquivo pode conter informações importantes para validação de acesso ao mesmo, como o histórico de sessão, e sem este controle pode ocasionar na inoperabilidade do *Scraping*. É possível gerenciar *cookies* utilizando a "Net::HTTP*"*, mas esta seria uma tarefa muito trabalhosa, quando é necessário e recomendado utilizar alguma biblioteca, que implemente este gerenciamento e outras funcionalidades encontradas em um navegador padrão.

Na linguagem Ruby existe uma *gem* (biblioteca) especial para execução desta tarefa, como é o caso da biblioteca "Mechanize*",* como não é uma biblioteca padrão, é necessário realizar o download da mesma, como especificado na Figura 7. Esta biblioteca implementa

um navegador com gerenciamento de sessão (armazenamento de cookies) entre diversos outros métodos automaticamente.

Figura 7 - Representação da instalação da gem Mechanize.

Fonte: Próprio Autor

A versão Ruby da biblioteca Mechanize, foi desenvolvida por Michael Neumann, tendo como base o modulo WWW::Mechanize, da linguagem de programação Pearl, implementada inicialmente por Andy Lester (LEE; 2010). Esta biblioteca permite que o programa (*Scraping*) interaja com um determinado site, e armazene suas informações, o que facilita na ação posterior, de extração de dados.

A outra atividade que um *Scraping* deve desempenhar, como vista nesta pesquisa, é a tarefa de processar documentos HTML em busca dos dados solicitados. É difícil de se executar utilizando apenas funções básicas de processamento de texto como buscas de palavras, substituições ou *loops* de leitura de caracteres. Para isto, existe uma funcionalidade presente neste software chamado de *parsing* (implementado pela biblioteca Nogokiri, em Ruby e inclusa no pacote de instalação da *gem* Mechanize), permitindo encontrar elementos de páginas HTML e XML pelos seletores de *Cascading Style Sheets* (CSS), por meio da identificação das estruturas dos documentos HTML, como a hierarquia de suas *tags* (Hunter Powers, 2013). Outra forma de realizar a identificação dos elementos nestas páginas são as expressões regulares, utilizadas para obter-se conteúdos que seguem certo padrão em um texto, e para a filtragem dos conteúdos extraídos.

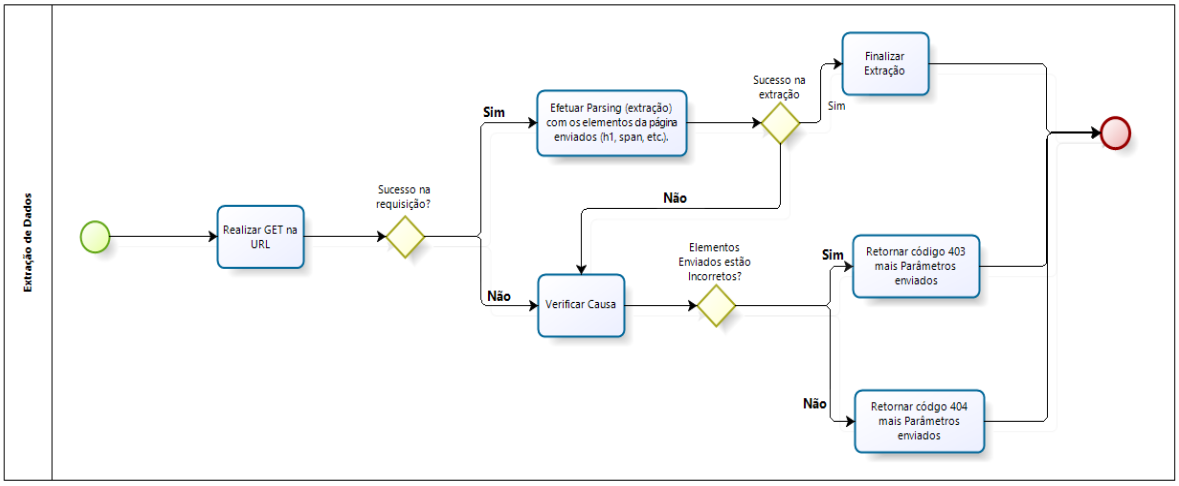
Ao efetuar a utilização da biblioteca Nogokiri, ou qualquer outra funcionalidade que permite o mesmo processamento, é de suma importância compreender as estruturas das páginas Web, ou seja, como é que os dados estão dispostos nela, na maioria das vezes eles encontram-se em estruturas HTML, que para a apresentação dos dados são utilizados seus marcadores (*tags*), como *div, h1, table, article,* entre diversos outros.

No entendimento desta estrutura, é um pré-requisito para o serviço em questão, que seja realizada uma análise, antes de desempenhar o acionamento do *Scraping*, e para isso existem ferramentas que podem facilitar esta tarefa, como *add-ons*, encontrados em navegadores como Google Chrome e Mozilla Firefox e outros, é possível o estudo de tal modo a facilitar a compreensão desta estrutura. As ferramentas comumente utilizadas são o Developer Tools e o Firebug, que se integram de tal maneira aos navegadores e permitem a visualização, edição, debug e monitoramento de HTML, CSS, JavaScript além do tráfego de

rede em tempo de carregamento de qualquer página Web, por meio da funcionalidade de inspecionar elementos, permite identificar e entender as estruturas das páginas.

Com base em todos esses quesitos, as atividades exercidas pelo *Scraping* desenvolvido no trabalho, pode ser analisado na Figura 8, em que o processo de extração de dados inicia-se com realização do GET na URL para que o download da página seja realizado, caso neste momento ocorra algum erro, é dado início a uma verificação, descrita anteriormente, com a ocorrência de sucesso na extração, a ação do robô é então finalizada.

Figura 8 - Representação do processo do subprocesso de Extração de dados.



Fonte: Próprio Autor.

Assim, com o termino dessa fase, a API retornará os dados no formato JSON ao Cliente, o qual realizará os seus próprios procedimentos, como análise, armazenamento, comparações, entre outras atividades que o mesmo pode julgar ser eficiente para o seu processo de negócio.

4.2 Especificação do Serviço Web Cliente

Este serviço tem por objetivo demostrar a utilização da API, e está divido em duas principais áreas, as quais tem por objetivo simular a construção e a execução de um agente de extração. As áreas são, Área de Parametrização do Web Scraping e Área de Recuperação das Informações, cada uma possui o seu próprio objetivo e público alvo, a fim de possibilitar a captação de diversos dados em inúmeros domínios Web, para que assim sejam usados para o enriquecimento de grandes bases de dados, oferecendo mais informações a serem analisadas.

Como afirmado neste capítulo, a linguagem de programação Ruby on Rails permite o desenvolvimento de aplicações Web de modo rápido e conciso, devido a mesma ser baseada

no modelo ActiveRecord, que utiliza as convenções de nomes para realizar o mapeamento dos os objetos do banco de dados, por meio de uma classe Object-Relational Mapping (ORM, um conjunto de técnicas para a transformação entre os modelos orientados a objetos e relacional), que com base em suas regras permite que as configurações sejam mínimas (CAELUM, 2014). Porem como o banco de dados usados nesta aplicação não seguem as características tradicionais, abstraídas pelo modelo ActiveRecord, o mesmo deve ser descartado no momento de sua criação, para que a aplicação não seja criada com base em bancos de dados relacionais.

No desenvolvimento deste serviço optou-se pela utilização de um banco de dados não-relacional, como MongoDB, a opção por este modelo, deu-se por meio das suas facilidades de integração, escalabilidade, tempo de resposta, e o fato do mesmo ser muito utilizado no cenário de Big Data. Para integração do banco de dados e do Framework Rails é necessário instalar uma nova *gem*, *mongo_mapper,* possibilitando a manipulação dos objetos e seus dados.

O objeto "Scraper" criado no MongoDB, possui as seguintes chaves (keys) "id" (permite a identificação de um agente), "*name*" contem a nomenclatura ideal que descreve o S*craping*, "*url*" e "*elements*" os quais serão usados posteriormente para a submissão a API, por meio deste é possível manipular os dados presentes no agente de extração. Como forma de fornecer uma melhor compreensão sobre o objeto criado a Tabela 4 traz a representação de suas chaves e a sua descrição.

Tabela 4 - Representação do objeto Scraper no MongoDB.

Chave	Descrição
Id	Representa o identificador único sobre o agente criado, assim á a possibilita a criação de índices, uma forma de permanecer algumas regras presentes em modelos tradicionais, e que pode ser útil neste modelo.
Name	Representa o nome do agente de extração assim é possível que o usuário possa identifica-lo.
Url	Representa um dos elementos necessários para a captação de dados, o mais importante a URL de acesso.
Elements	Representa os elementos presentes na página HTML, assim finalizando as regras necessárias para o Web Scraping

A criação do objeto "Scraper", permite aos usuários abstrair de forma simples as informações relevantes para o estabelecimento das regras do robô de busca, a área de

parametrização possibilita está interação e manipulação dos dados. A ideia primaria para estabelecimento desta área, é de que seu uso especifico a especialistas de dados pois somente eles teriam os conhecimentos necessários para o estabelecimento das regras, como observado na Figura 5, podem inferir uma análise das fontes para extração.

A área que corresponde a recuperação das informações tem por finalidade somente a execução dos *Scrapings* criados, os dados cadastrados anteriormente são enviadas para a API, por meio de uma URL (“/scrapers/execute/5444250c720f2a118c000002”), sendo acessado pelo identificador que compreende a sua chave “id” criada na aplicação (5444250c720f2a118c000002), tendo por objetivo atender a necessidade de usuários finais que dependem da extração dessas informações, pois assim, com as informações já previamente cadastradas, o usuário final desta aplicação não possui a necessidade de saber quais são as regras utilizadas para a efetividade da ação de extração de dados.

4.3 Trabalhos Correlatos

O trabalho foi desenvolvido com base nas pesquisas realizadas, e durante este processo pode-se observar trabalhos que assemelham-se a este. O primeiro analisado foi também uma API de extração de dados, que utilizava um *Scraping*, para realizar esta ação, em páginas HTML, o software em questão possui algumas características em comum, porém o usuário não faz o envio de elementos contidos nas páginas como a estrutura hierárquica do HTML da página e sim o envio somente da URL que pretende-se extrair e palavras chaves as quais simbolizam como a captura dos dados pode ser realizada, desta forma permitindo que mais funcionalidades sejam exploradas e informações colhidas (ALCHEMYAPI, 2014).

O software foi desenvolvido pela empresa, AlchemyAPI, que com base em suas pesquisas sobre os avanços na Inteligência Artificial (AI) estão desenvolvendo uma plataforma em nuvem que permite aos desenvolvedores construírem rapidamente aplicações que permitam atingir novos conhecimentos e capacidades, por meio de aplicações de análise de texto avançados, visão computacional e serviços de armazenamento de dados para grandes volumes de dados não estruturados.

Algumas soluções apresentadas por esta empresa auxiliaram a estabelecer o cumprimento da criação da plataforma de extração, que também possui o objetivo a auxiliar profissionais da área a capacidade de discernir de forma mais eficiente os processos posteriores a extração, como armazenagem, analise e a agregação de valores os dados.

O segundo trabalho correlato possuiu como objetivo o estudo de técnicas de extração

com base em dados semiestruturados, esta dissertação de mestrado desenvolvida por Mendonça (2003), intitulada “Extração Resiliente de Dados RDF a partir de Fontes Dinâmicas em Linguagem de Marcação”, foi utilizada como referência para este trabalho na elaboração dos estudos inicias sobre as técnicas de extração de dados. A dissertação desenvolvida por este autor, permitiu a compreensão da importância da extração de dados pela perspectiva de sua manipulação posterior, a fim de auxiliar na elaboração de estratégicas de extração de fontes dinâmicas na Web (MENDONÇA; 2003).

Os dois trabalhos apresentados possibilitaram o planejamento dos estudos e da estrutura desenvolvida neste trabalho de conclusão de curso agregando os conhecimentos essências para o mesmo, e por fim gerando a plataforma de extração e recuperação na Web no contexto de Big Data.

4.4 Testes Efetuados

Na composição deste trabalho a aplicação que compreende ao serviço Cliente, foi desenvolvida com o intuito de demostrar as funcionalidades apresentadas na API, por meio de sua Área de Recuperação das Informações, que permitiu a verificação e validação de alguns dos principais requisitos para o estabelecimento da integração entre as duas aplicações, como o envio de parâmetros validos e inválidos, o tempo de resposta para cada solicitação e a verificação da resposta e seu formato de retorno. Assim resultando é visualizado na Tabela 5, com os respectivos testes realizados:

- O primeiro teste, foi enviado todos os parâmetros aceitos pela API, porém a hierarquia de *tags* enviadas na consulta não encontrava-se no site, o qual assim retornou a seguinte mensagem “*403 - The data does not exist in the source site's database.*”;
- O segundo teste realizado, contou com apenas envio de um dos parâmetros validos, e a consulta apresentou o código “*402 - Invalid parameters were sent.*”, que corresponde a parâmetros inválidos;
- O terceiro teste realizado teve como objetivo o envio de todos os dados necessários, a fim de, retornar o resultado referente a extração, porém devido a um erro inesperado ocasionado pela falha de conexão, o código retornado neste caso é “*404 - Error Unexpected*”; e
- O quarto teste efetuado teve como objetivo o envio de todos os dados

necessários para que ocorre-se a extração e seu código de resposta foi “200 – Success action.”, mais o resultado da consulta.

Tabela 5 - Representação dos Testes realizados.

Nº	Parâmetros de Entrada	Código de Resposta
1	{“url”:http://g1.globo.com/politica/eleicoes/2014/noticia/2014/10/dilma-e-aecio-se-atacam-no-primeiro-debate-do-2-turno-na-tv.html, “elements”: “span, h1” }	*403 - The data does not exist in the source site's database.*

2	{
	“url”:http://g1.globo.com/politica/eleicoes/2014/noticia/2014/10/dilma-e-aecio-se-atacam-no-primeiro-debate-do-2-turno-na-tv.html,
	“elements”: “”
	}
		402 - Invalid parameters were sent.

3	{
	“url”:http://g1.globo.com/politica/eleicoes/2014/noticia/2014/10/dilma-e-aecio-se-atacam-no-primeiro-debate-do-2-turno-na-tv.html,
	“elements”: “h1”
	}
		404 - Error Unexpected.

4	{
	“url”:http://g1.globo.com/politica/eleicoes/2014/noticia/2014/10/dilma-e-aecio-se-atacam-no-primeiro-debate-do-2-turno-na-tv.html,
	“elements”: “h1”
	}
		200 – Success action, mais o Resultado da extração.

Os testes realizados tiveram como objetivo demonstrar as funcionalidades das consultas realizadas na API, e quais tipos de resultados poderiam ser retornados em uma consulta, o que pode ser observado na Tabela 4. Os resultados para os testes, obtiveram êxito e permitiram assegurar o esperado para cada parâmetro de consultas.

4.5 Considerações Finais do Capítulo

O processo que levou a definição da arquitetura da plataforma de extração e recuperação de dados, foi estabelecida no intuito de promover os estudos referente as áreas abordadas, além de demostrar a utilização de técnicas como Web Scraping para a captação de dados no ambiente Web. Diante desta motivação a sua abstração na criação de um serviço como uma API baseada na arquitetura REST, foi implementada a fim de possibilitar o seu uso por meio de outros serviços – como o desenvolvido também neste trabalho – deste modo ampliando seus conceitos nos ambientes acadêmicos e empresariais.

5 RESULTADOS E CONTRIBUIÇÕES

Com o desenvolvimento deste trabalho, realizado por intermédio das pesquisas sobre as técnicas de extração de dados em páginas HTML, obteve-se a criação de uma arquitetura de serviço baseada nessas características. Neste contexto, os Web Scrapings, podem proporcionar o desenvolvimento de soluções que atendam diversas necessidades em relação as tecnologias de extração e recuperação de informações no cenário de Big Data. Este capítulo apresenta os resultados colhidos com o desenvolvimento das aplicações vigentes, demostrando as formas que a mesma realiza as extrações e como o resultado é retornado para o usuário, e por fim apresenta como os dados disponíveis neste ambiente podem ser de grande importância para nichos de negócios que necessitam dessas informações.

5.1 Resultados da Plataforma

A extração de dados realizada pelo *Scraping* possui algumas etapas como especificadas no capítulo 4, que para a sua realização, o usuário deve analisar a fonte a qual deseja-se os dados, para o envio correto dos elementos, possibilitando a sua captura. A Figura 9, apresenta uma página a qual será feita esta ação, na página os elementos enviados devem ser, a sua URL e as *tags* que contenham as informações.

Figura 9 - Fonte para a extração de dados.

Fonte: Site G1 política.

O conteúdo selecionado na Figura 9 compreende-se a título da matéria, com a seguinte informação:

> “Dilma e Aécio respondem a indecisos e mantêm acusações no debate final. Confronto na TV Globo foi ultimo antes do segundo turno da eleição. Eleitores indecisos participam fazendo perguntas aos presidenciáveis.” (Política, G1; 2014).

Para efetuar a extração desta informação, deve-se analisar a estrutura HTML em que a mesma encontra-se, utilizando *add-on* Developer Tools disponibilizado no browser Google Chrome é possível está verificação, a qual pode-se analisar na Figura 10.

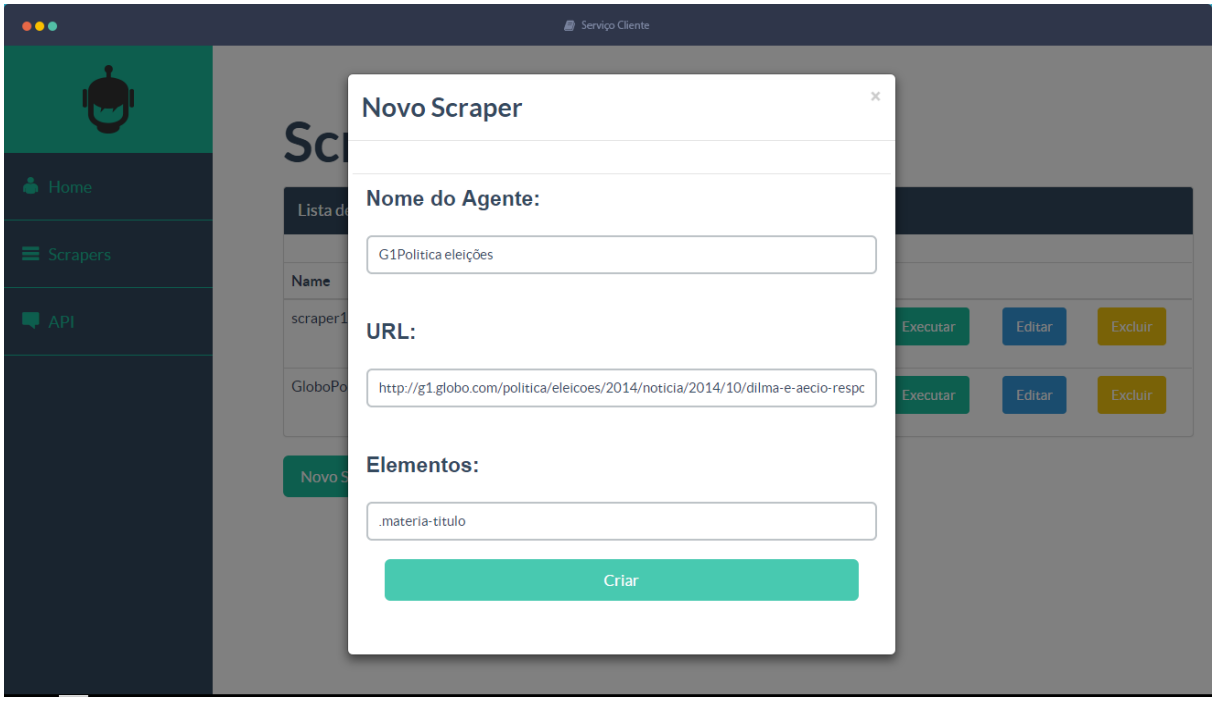
Figura 10 - Estrutura HTML, em que encontra-se os dados para extração.

```
<!-- google_ad_section_start -->
▼<div class="materia-titulo">
  <h1 class="entry-title">Dilma e Aécio respondem a indecisos e mantêm acusações no debate final</h1>
  ▼<h2>
    "Confronto na TV Globo foi o último antes do segundo turno da eleição."
    <br>
    "Eleitores indecisos participaram, fazendo perguntas aos presidenciáveis."
  </h2>
</div>
```

Fonte: Developer Tools, site G1 política.

O conteúdo está contido em uma “<*div*>” que possui uma classe CSS, “.materia-titulo” e com base nela a extração pode ser realizada. O segundo passo agora utilizando o serviço Cliente desenvolvido no trabalho, é cadastrar as informações inerentes a extração, que serão enviados para a API, e a Área de Parametrização do Web Scraping permite que tal ação seja desempenhada, como demostrado na Figura 11, por meio de seu formulário.

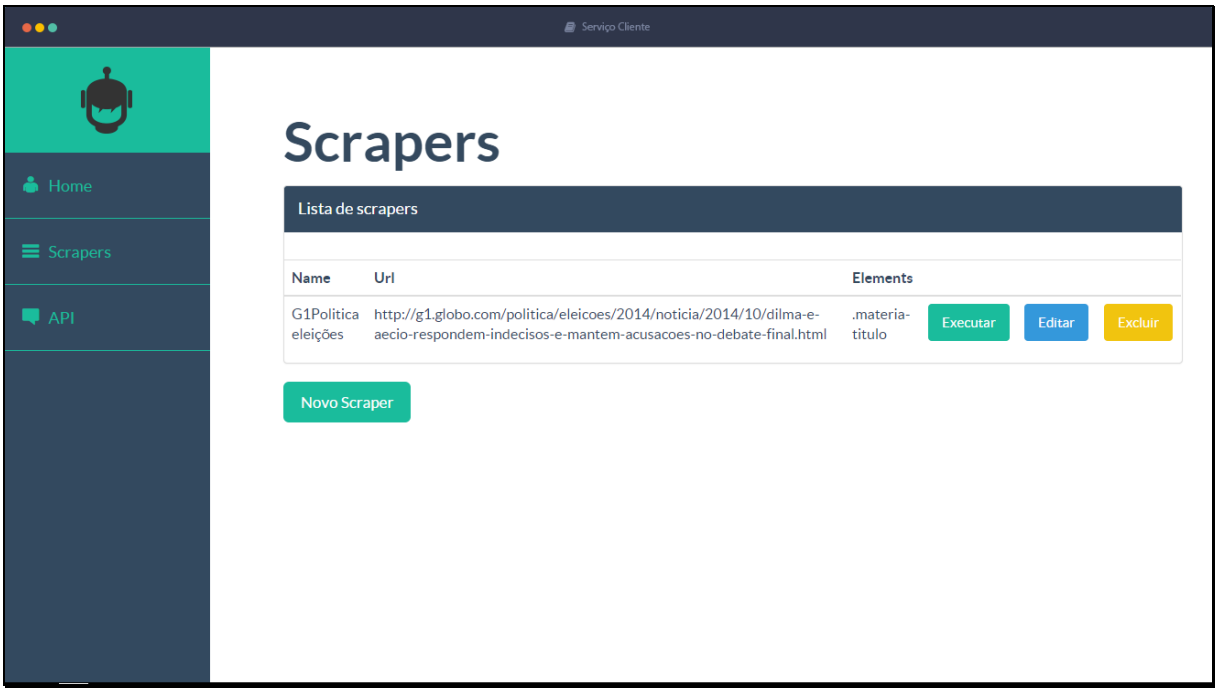
Figura 11 - Formulário de cadastro da Área de Parametrização.



Fonte: Próprio Autor.

Após esta etapa as informações são armazenadas no MongoDB, as quais poderão ser posteriormente manipuladas, e com isso, pode-se efetuar a execução do *Scraping* criado, ativando assim o recurso de extração (API), a Figura 12 apresenta a visualização dos parâmetros cadastrados.

Figura 12 - Visualização da lista de parâmetros cadastrados.



Fonte: Próprio Autor.

Com a integração das duas aplicações, os dados podem ser enviados ao executar o *Scraping* o qual é realizado no botão "Executar" e assim acionado a API. Com isso os dados são retornados no formato JSON, como expostos na Figura 13.

Figura 13 - JSON retornado na consulta.


```
{
    "time": "2014-10-25T20:52:01Z",
    "params": {
        "url": "
http://g1.globo.com/politica/eleicoes/2014/noticia/2014/10/dilma-e-aecio-respondem-indecisos-e-mantem-acusacoes-no-debate-final.html
",
        "elements": ".materia-titulo"
    },
    "data": [
        "Dilma e Aécio respondem a indecisos e mantêm acusações no debate final Confronto na TV Globo foi o último antes do segundo turno da eleição.Eleitores indecisos participaram, fazendo perguntas aos presidenciáveis."
    ],
    "message": "200 - Successful action."
}
```


Fonte: Próprio autor.

O processo apresentado até agora demostrou a utilização dos dois serviços desenvolvidos, a fim de, confirmar o funcionamento dos mesmos, as informações extraídas

são observadas na chave "*data*", o qual retorna um Array com os dados, as mesmas informações só serão garantidas se o usuário for capaz de enviar os elementos corretamente. No processo descrito as informações repassadas para a API são validadas, como abordado nos fluxos desenhados nas Figuras 6 e 8, caso ocorra alguma falha durante este processo são retornados os códigos descritos anteriormente, para demonstrar isso a Figura 14, apresenta como o JSON é retornado quando o elemento enviado para a extração não é encontrado na página solicitada.

Figura 14 - JSON retornado quando não é encontrado o elemento solicitado.


```
{
    "time": "2014-10-25T21:37:55Z",
    "params": {
        "url": "
http://g1.globo.com/politica/eleicoes/2014/noticia/2014/10/dilma-e-aecio-respondem-indecisos-e-mantem-acusacoes-no-debate-final.html
",
        "elements": ".materia-titul"
    },
    "data": {
        "code": "403 - The data does not exist in the source site's database."
    },
    "message": "200 - Successful action."
}
```


Fonte: Próprio autor.

Assim, como em caso de sucesso todos os dados são retornados, inclusive a chave "*message*" com o código de sucesso na requisição 200, porém na chave "*data*" é retornado apenas o código de falha 403, o qual condiz que "*The data does not exist in the source site's database.*" (Os dados não existem na base de dados do local de origem), significando que a análise prévia da página não foi realizada corretamente, o que resultou nesta falha. Quando a falha é mais grave como a página encontra-se inoperante ou indisponível, ou ainda ocorreu o bloqueio de acesso a mesma, o retorno para o serviço cliente tem como base o seguinte código, 404 com a mensagem de "Erro Unexpected." (Erro inesperado), apresentado na Figura 15.

Figura 15 - JSON para erro inesperado.


```
{
    "code": "404 - Error Unexpected."
}
```


Fonte: Próprio Autor.

Afim de estabelecer o exercício de extração de dados de qualquer página HTML, deu-se por meio da generalização dos métodos de extração resultando em um Web Scraping parametrizado, e as regras estão sobre controle do usuário. Desta forma a aplicação pode

atingir um nível maior de fontes capazes de obter-se os dados, mesmo sendo limitada a regras estabelecidas, pois como já apresentado a extração se faz somente com o uso do método de requisição GET, o qual geralmente compreende ao primeiro acesso a página, efetuando o seu "download" e em seguida a extração da mesma.

5.2 Restrições da API

No processo de criação da principal aplicação deste trabalho, a qual refere-se a API, medidas de confiabilidade em seus resultados foram implementadas, e a cada caso especifico encontrado foi definido um código de requisição como apresentado anteriormente na Tabela 3, assim possibilita que os serviços a utilizá-la possam identificar e tratar tais resultados.

Os códigos definidos representam as possíveis limitações na própria API, pois para enfatizar seu uso é necessário que suas especificações sejam cumpridas corretamente, como o envio dos parâmetros *url* e *elements*, sem os quais não será possível efetuar a extração dos dados corretamente, tendo em vista que antes do envio dessas informações é essencial a análise do padrão em que os dados estão dispostos nas páginas, ou seja, a estrutura do HTML.

Para um esclarecimento assertivo das limitações do serviço (API) neste trabalho optou-se pelo desenvolvimento simplificado de suas técnicas de extração, visto que também foi criado um outro serviço Web que a consume, como observado no subcapítulo 5.1, onde é demostrado o passo a passo para a utilização da plataforma, entende-se que a extração é efetiva se o envio dos parâmetros for correto. A grande problemática encontra-se na fonte de dados que deve-se efetuar está ação, pois como já escrito os métodos de extração utilizados na constituição do Web Scraping parametrizado, fornecem uma resposta baseada nos elementos que forem encontrados nas páginas.

A principal fonte de dados que o serviço extrai é apresentada como sendo uma estrutura HTML, por ser a base de todos os sites presente na Web, devido a sua padronização ser especificada pelo órgão regulamentador W3C, o qual valida todas estruturas de marcações. Com o avanço tecnológico dos Browsers tais como Google Chrome, Mozila Firefox entre outros, algumas deficiências como a má formação das tags no processo de criação de um documento Web, como por exemplo a tag a qual demarca os dados não foi fechada corretamente, estes erros são supridos pelos mesmos, o que torna-se quase impossível notar-se somente com uma análise visual da estrutura.

O Web Scraping desenvolvido neste trabalho realiza a extração de dados com um simples *parsing* HTML com base no elemento enviado como parâmetro (na chave

‘*elements’*), caso a fonte de dado escolhida tenha falhas como já relatada a provável resposta será ‘404 – Error Unexpected.’ no JSON, pois os métodos de extração efetuam a captura dos dados por meio da busca desses elementos no corpo (body) do documento.

Uma restrição quanto a uma das especificações da API, está relacionada com o parâmetro ‘*url*’, uma URL é composta geralmente por um esquema ou protocolo como o HTTP, um domínio qual remete ao servidor que encontra-se, um caminho que especifica um recurso a ser acessado entre outras características encontradas na mesma (RICHARDSON; 2007). Para que o acesso a página requisitada seja efetivado é necessário que este parâmetro esteja correto, ou seja, possua estas características, o qual deve ser validado pelo serviço que está consumindo a API, caso contrário o retorno será ‘402 - Invalid parameters were sent.’, pois a mesma somente encarregará de efetivar o acesso a URL.

Portanto a aplicação possui restrições quanto ao envio dos parâmetros para a efetuar a extração dos dados de acordo com as especificações solicitadas pelo serviço ou usuário que a consome. Outro ponto importante quanto a maneira em que são capturadas as informações dos domínios requeridos, está no fato de que o Web Scraping desenvolvido efetua somente uma requisição por meio do método de requisição GET o qual permite somente a navegação na página que corresponde a este método.

Uma limitação que pode ser encontrada na API está relacionada a falta de funcionalidades que atendam a todas as necessidades como por exemplo, caso seja enviado uma URL que necessita da validação de demais parâmetros no momento da requisição, no qual é comumente utilizado o método de requisição POST, este serviço não suportaria tais necessidades e retornaria uma das mensagens de erro, com relação a este requisito tornou-se um provável trabalho futuro em relação ao aprimoramento dos métodos de extração utilizados.

Contudo essas restrições encontradas no envio de parâmetros e nos métodos que efetuam a extração de dados, revelam que, assim como todo software desenvolvido, a necessidade do aprimoramento e aprofundamento em pesquisas são de grande importância para o cenário acadêmico cientifico e corporativo.

5.3 Considerações Finais do Capítulo

Os estudos referentes as áreas de extração de dados em diversos domínios, como da Internet, realizada neste trabalho, remetem a capacidade de empresas e organizações agregarem mais valores aos dados já existentes, proporcionando a geração de conhecimentos e a sua capacidade de ampliar as perspectivas sobre a inteligência de negócios.

O aumento de dados torna-se cada vez mais necessário a inserção de discernimento nas ferramentas que apoiam a tomada de decisão, de modo que exista a flexibilidade em administrar grandes volumes de dados de maneira eficaz e eficiente. Neste contexto, os estudos das técnicas e métodos, como as que foram apresentadas, os Scrapings, podem sim proporcionar o desenvolvimento de ferramentas que atendam esses requisitos, visando a melhor forma de capturar dados de diversos domínios e contextos.

As contribuições deste trabalho visaram demostrar a importância da utilização destes softwares para a captura das informações em domínios como a Web, por meio da apresentação de suas técnicas, sendo o primeiro passo para o estabelecimento de uma arquitetura de serviços que forneçam as ferramentas necessárias para a captura e posteriormente armazenagem, e análise dos dados.

CONCLUSÃO

Destacou-se que a área de extração de dados pode ser uma grande aliada nos esforços para estabelecer novas fontes heterogêneas para a captação dos dados, por intermédio de suas tecnologias que são capazes de realizar esta ação, como o caso da utilizada para este trabalho, o Web Scraping.

Indubitavelmente a área de extração de dados oferece recursos essenciais para a captura-los, que em tecnologias inseridas no contexto de Big Data é de suma importância o fortalecimento de suas bases dados de modo a suprirem e suportarem os mais variados formatos como estruturados, semiestruturados e não estruturados disponíveis em diversos domínios, a fim de inferir-se uma análise com base nos mesmos.

A grande massa de dados gerados hoje em diversos cenários como os presentes em portais de notícias, mídias sociais, blogs, fóruns, sites governamentais, entre outros apresentados durante toda pesquisa, tem grande importância na consolidação de uma plataforma de Big Data, visto que, na necessidade de instaura-la torna-se cada vez mais essencial a importância de uma análise ampla de diferentes fontes, proporcionando uma visão mais sofisticada sobre diversos contextos.

Neste trabalho utiliza-se um Web Scraping, para a extração de fontes como já citadas, pois esses softwares possuem melhores recursos de inteligência em relação aos outros robôs de busca, e a sua principal característica é que as informações extraídas por eles devem ser tratadas, assim podendo facilitar em futuras análises e manipulações dos mesmos.

Portanto o desenvolvimento da pesquisa optou-se na especialização de um robô de busca, o qual suas regras estão sobre controle de um usuário especialista na área de dados, proporcionando ao mesmo a capacidade e a facilidade de instaurar a captação de dados. Este objetivo foi atingido com o desenvolvimento de uma REST API que abstrai um Web Scraping, proporcionando a capacidade de integração com diversas outras aplicações.

O desenvolvimento de uma aplicação que engloba um agente de extração e baseado em uma arquitetura de serviço REST permite a compreensão do negócio que pretende-se auxiliar, que no caso é servir a especialistas da área de dados de modo a estabelecer-se em uma plataforma de Big Data na tarefa de extração e recuperação de dados. Com a sua criação, optou-se em integrar com outro serviço Web, produzido neste trabalho também, e que demostrou os recursos, especificações e funcionalidade da API.

A divisão de responsabilidade é o verdadeiro benefício quanto ao desenvolvimento de softwares que seguem a arquitetura REST, pois suas vantagens estão na rapidez, baixo custo na sua produção e grande capacidade de escala-las de modo a permitir o uso por vários outros serviços.

No estabelecimento de uma plataforma de Big Data seguir as suas características essenciais é um pré-requisito, o qual diz respeito aos três principais ‘Vs’ (Variedade, Volume e Velocidade), pois cada um expressa suas próprias diretrizes para melhor atender as necessidades de trabalhar com grandes quantidades de dados de diversos formatos, e tecnologias inseridas nos contextos de NoSQL, como o MongoDB possibilitam a manipulação de forma livre de algumas regras tradicionais favorecendo no desenvolvimento de aplicações, como o outro serviço Web implementado durante a pesquisa, onde são armazenados os dados referentes as regras do agentes de extração.

Assim pode-se enfatizar em uma plataforma de extração e recuperação de dados, que pode vir a fortalecer a compreensão de cenários ou ambientes de negócios que necessitam de apoio na criação de softwares desta alçada. Pois uma vez observado estas necessidades, mais soluções cabíveis em relação as empresas e organizações são agregadas neste contexto, como por exemplo, a implementação de novos serviços que suportam as características especificas em relação a aplicação criada neste trabalho.

Estes resultados observados com as pesquisas sobre as áreas apresentadas e com a concretização das mesmas em conjunto a arquitetura de plataforma desenvolvida, demostrou as necessidades da realização de pesquisas e estudos referente a área de extração e recuperação de dados favorecendo trabalhos futuros.

Trabalhos Futuros

Como trabalhos futuros, para o fortalecimento das pesquisas e das soluções desenvolvidas neste trabalho, são propostos:

- O fortalecimento dos métodos parametrizados do Web Scraping, para que o mesmo possa realizar extrações cada fez mais complexas;
- A disponibilização da API em um serviço em nuvem, como uma Platform as a Service (PaaS) enfatizando seu uso por meio de outros softwares;
- A possibilidade de estabelecer-se em uma plataforma de Big Data para o auxílio da extração de dados;

- O aprofundamento em pesquisas que explorem mais recursos disponíveis nas áreas e tecnologias utilizadas durante este trabalho.

REFERÊNCIAS

ALMEIDA, Ricardo; BRITO, Parcilene. Utilização da Classe de Banco de Dados NOSQL como Solução para Manipulação de Diversas Estruturas de Dados. Disponível em: http://www.bandalerda.com.br/wp-content/uploads/2012/10/Utilizacao_da_Classe_de_Banco_de_-Dados_NOSQL_como_Solucao_para_Manipulacao_de_Diversas_Estruturas_de_Dados.pdf. Acessado em: agosto de 2014.

BAMFORD, James. Três dias com Edward Snowden, o homem mais procurado do mundo. Disponível em: http://gq.globo.com/Prazeres/Poder/noticia/2014/10/tres-dias-com-edward-snowden-o-homem-mais-procurado-do-mundo.html. Acessado em: outubro de 2014.

BAUMGARTNER, Robert, et. al. Web Data Extraction for Business Intelligence: the Lixto Approach. Disponível em: http://pdf.aminer.org/000/069/407/web_data_extraction_for_business_intelligence_the_lixto_approach.pdf. Acessado em: março de 2014.

BRIN, Sergey. Extracting Patterns and Relations from the World Wide Web. Disponível em: http://ilpubs.stanford.edu:8090/421/1/1999-65.pdf. Acessado em: abril de 2014.

BRITO, W. Ricardo. Bancos de Dados NoSQL x SGBDs Relacionais: Análise Comparativa. Disponível em: http://infobrasil.inf.br/userfiles/27-05-S4-168840-Banco%20de%20Dados%20NoSQL.pdf. Acessado em: agosto de 2014.

CAVALCANTI, José. Tendências de Inovação para a Tecnologia de Big Data. Disponível em: http://pt.slideshare.net/cictec/tendencias-de-inovaes-para-atecnologia-de-big-data. Acessado em: abril de 2014.

COURTNEY, Martin. The Larging-UP of Big Data. Disponível em: http://buyukverienstitusu.com/s/1870/i/The_Larging_Up_Of_Big_Data.pdf. Acessado em: abril de 2014

D'ANDRÉIA, Carlos. Estratégia de produção e organização de informações na Web: conceitos para a análise de documentos na Internet. Disponível em: https://www.scielo.br/pdf/civ35n3/v35n3a04. Acessado em: abril de 2014.

DEAN, Jeffrey; GHEMAWAT, Sanjay. MapReduce: Simplified Data Processing on Large Clusters. Disponível em:

http://static.googleusercontent.com/media/research.google.com.com/pt-BR//archive/mapreduce-osdi04.pdf. Acessado em: setembro 2014.

DEITEL, H.M.; DEITEL, P.J; NIETO, T.R. Big Data: A Revolution that Will Transform how We Live, Work, and Think. Bookman. 2ª ed., 2003.

FERRARA, Emilio; CATANESE, A. Salvatore; MEO, Pasquale; FIUMARA, Giacomo; PROVETTI, Alessandro. Crawling Facebook for Social Network Analysis Purposes. Disponível em: http://arxiv.org/pdf/1105.6307.pdf. Acessado em: abril de 2014.

FERRARA, Emilo; MEO, Pasquale; FIUMARA, Giacomo; BAUMGARTNER, Robert. Web Data Extraction, Applications and Techniques: A survey. Disponível em: http://arxiv.org/pdf/1207.0246v3.pdf. Acessado em: junho de 2014.

FILHO, Antônio; DELGADO, Maria. A sobrecarga de informação na era da Internet. Disponível em: https://www.espacoacademico.com.br/023/23amsf.htm. Acessado em: Abril de 2014.

GERHARD, Bill; GRIFFIN, Kate; KLEMANN, Roland. Unlocking Value in the Fragmented World of Big Data Analytics: How Information Infomediaries Will Create a New Data Ecosystem. Disponível em: https://www.cisco.com/web/about/ac79/docs/sp/Information-Infomediaries.pdf. Acessado em: maio de 2014.

GRIHMAN, Raphael. Information Extraction: Tecniques and Chalenge. Disponível em: http://www.ru.is/faculty/hrafn/Papers/grishman97information.pdf. Acessado em: abril de 2014.

IMPERVA. Detecting and Blocking Site Scraping Attacks. Disponível em: http://www.imperva.com/docs/WP_Detecting_and_Blocking_Site_Scraping_Attacks.pdf. Acessado em: março de 2014.

KAISLER, S.; ARMOUR, F. et al. Big data: Issues and challenges moving forward. System Sciences (HICSS), 2013 46th Hawaii International Conference, p. 995 – 1004, em IEEE 2013.

KAYAALP, Mehmet; OZYER, Tanzel; ÖZYER, Sibel. A Collaborative and Content Based Event Recommendation System Integrated With Data Collection Scrapers and Services at a Social Networking Site. Disponível em: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=5231918&url=http%3A%2F%2Fieeexplore.org%2Fxpls%2Fabs_all.jsp%3Farnumbers%3D5231918. Acessado em: maio de 2014.

KAKHANI, K.; KAKHANI, S.; BIRADAR. Research Issues in Big Data Analytics. Disponível em: http://www.ijaiem.org/volume2issue8/IJAIEM-2013-08-29-070.pdf. Acessado em: abril de 2014.

KALE, Mr. Swapnil; DANDGE, Sangram. Understand The Big Data Problems and Their Solutions Using Hadoop And Map-Reduce. Disponível em: http://www.ijaiem.org/volume3issue3/IJAIEM-2014-03-31-134.pdf. Acessado em: maio de 2014.

KATAL, A.; WAZID, M.; GOUDAR, R.H. Big Data: Issues, Challenges, Tools and Good Practices. Disponível em: http://ieeexplore.ieee.org/xpl/login.jsp?tp=&arnumber=6612229&url=http%3A%2F%2Fieeexplore.ieee.org%2Fxpls%2Fabs_all.jsp%3Farnumber%3D6612229. Acessado em: abril de 2014.

LAENDER H. F; et. al. A brief survey of web data extraction tools. Disponível em: http://www.sigmod.org/publications/sigmod-record/0206/laender-survey.pdf. Acessado em: março 2014.

LEAVITT, Neal. Will NoSQL Databases Live Up to Their Promise?. Disponível em: http://www.leavcom.com/pdf/NoSQL.pdf. Acessado em: maio 2014.

LEE, John. Mechanize. Disponível em: http://wwwsearch.sourceforge.net/mechanize/. Acessado em: setembro 2014.

MASNICK, Mike. Can Scraping Non-Infringing Content Become Copyright Infringement Because Of How Scrapers Work?. Disponível em: https://www.techdirt.com/articles/20090605/2228205147.shtml. Acessado em: abril de 2014.

MASSÉ, Mark. REST API Design Ruleboook. Oreilly and associate Series. Ilustrada ed. O'Reilly Midia, Inc. 2011.

MATOS, Pablo. Metodologia de Pré-processamento textual para Extração de Informação sobre Efeitos de Doenças em Artigos Científicos do Domínio Biomédico. Disponível em: http://gbd.dc.ufscar.br/downlod/files/courses/NonConventionalDB_2010/PabloMatos2010_Dissertacao.pdf. Acessado em: julho de 2014.

MCAFEE, Adrew e BRYNJOLFSSON, Erick. Big Data: The Mangment Revolution. Disponível em: http://hbr.org/2012/10/big-data-the-management-revolution/ar. Acessado em: maio de 2014.

MEDIA, O'Reilly. Big Data Now: 2012 Edition. 2 ed.: "O'Reilly Media, Inc." 2012.

MENDONÇA, Eduardo. Extração Resiliente de Dados RDF a partir de Fontes Dinâmicas em Linguagem de Marcação. Disponível em: http://www.livrosgratis.com.br/arquivos_livros/cp108124.pdf. Acessado em: fevereiro de 2014.

MONIRUZZAMAN, A B M; HOSSAINS, Syed. NoSQL Database: New Era of Databases for Big Data Analytics – Classification, Characteristics and Comparion. Disponível em: http://arxiv.org/pdf/1307.0191.pdf. Acessado em: agosto de 2014.

MOTTA, Vladimir. Verdade e mitos sobre Big Data. Disponível em: http://www.stoneage.com.br/index.php/pt/blog/170-verdaddes-e-mitos-sobre-o-big-data. Acessado em: junho de 2014.

NATSUI, Erica. Inteligencia Competitiva. Disponível em: https://www.ead.fea.usp.br/tcc/trabalhos/artigo_Erica Natsui.pdf. Acessado em: Abril de 2014.

O'REILLY, Radar Team. Big Data Now: Current Perspectives from O'Reilly Radar.ed. O'Reilly Media, Inc., 2011.

OREND, Kai. Analysis and Classification of NoSQL Databases and Evaluation of their Ability to Replace an Object-relational Persistence Layer. Forschungs- und Lehreinheit XIX: Software Engineering for Business Information System, 2010.

OHLHORST, J. Frank. Big Data Analytics: Turning Big Data into Big Money. ed. Ilustrada, John Wiley & Sons, 2012.

RICHARDSON, Leonard; RUBY, Sam. RESTful Web services. 1 ed. Mike Loukides, 2007.

RUSSOM, Philip. Managing Big Data. Disponível em: http://www.pentaho.com/sites/default/files/uploads/resources/tdwi_best_practices_report-_managing_big_data.pdf. Acessado em: abril 2014.

SADALAGE, J. Pramod; FOWLER, Martin. NoSQL Essencial: Um Guia Conciso para o Mundo Emergente da Persistência Poliglota. ed. Novatec Editora LTDA, 2013.

SAGIROGLU, Seref; SINANC, Duygu. Big data: A review. Em: Collaboration Technologies and Systems (CTS), 2013 Conferência Internacional em IEEE, 2013. p. 42-47.

SANGER, E. David; SCHMITT, Eric. Snowden Used Low-Cost Tool to Best N.S.A. Disponível em: http://www.nytimes.com/2014/02/09/us/snowden-used-low-cost-tool-to-best-nsa.html?hp&_r=2. Acessado em: fevereiro de 2014.

SILVA, F.A Eduardo; BARROS, A. Flávia; PRUDÊNCIO B.C, Ricardo. Uma Abordagem de Aprendizagem Híbrida para Extração de Informação em Texto Semi-Estruturados. Disponível em: http://www.lbd.dcc.ufmg.br/colecoes/enia/2005/047.pdf. Acessado em: junho de 2014.

SILVAN P., Akhil; JOHNS, Jisha; VENUGOPAL, Jayasurya. Big data Intelligence on Logistics. Disponível em: http://www.irdindia.in/journal_ijacte/pdf/vol3_iss1/11.pdf Acessado em: abril de 2014.

TESORE, Carlos. Big data e inteligência analítica. Disponível em: http://www.dci.com.br/dci-sp/big-data-e-inteligencia-analitica--id380907.html. Acessado em: abril de 2014.

VAILAYA, Aditya. What's All the Buzz Around 'Big Data?'. Disponível em: http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf. Acesso em: abril 2014.

VIERA, Marcos Rodrigues, et al. Bancos de Dados NoSQL: conceitos, ferramentas, linguagens e estudos de casos no contexto de Big Data. Disponível em: http://data.ime.usp.br/sbbd2012/artigos/pdfs/sbbd_min_01.pdf. Acesso em: abril 2014.

W3C, World Wide Web Consortimum. Extensible Markup Language (XML). Disponível em: http://www.w3c.org/XML/. Acessado em: maio 2014.

WARDEN, Pete. Big Data Glossary. Disponível em: http://dl.e-book-free.com/2013/07/big_data_glossary.pdf. O'Reilly Media, Inc. 2011.

WYLIE, Brian; et al. Using NoSQL Databases for Streaming Network Analysis. Disponível em: http://www.cs.sandia.gov/~dmdunla/publication/WyDuDaBa12.pdf. Acessado em: setembro 2014.

ZHAO, Hongkun. Autmomatic Wrapper Generation for the Extraction of Search Result Records. State University of New York at Binghamton, Computer Science. ProQuest, 2007.

ZHAO, Hongkun; et al. Fully Automatic Wrapper Generation for Search Engines. Disponível em: http://cs.binghamton.edu/~meng/pub.d/p458-zha0.pdf. Acesso em: junho 2014.

ZIKOPOULOS, Paul C., et al. Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data. 1.ed. New York, NY. McGraw-Hill, 2011.