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

Luan Silveira Pontolio, *Centro de Inovação BVTec, Boa Vista Serviços*

***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.**

***Index Terms* – Big Data; Data Extraction; Web Scraping;**

I. 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 [1].

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 [2]. 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.

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.

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.

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. A pesquisa visa ainda 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 foi 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.

II. BIG DATA

Cada vez mais, as organizações enfrentam este crescente aumento na riqueza de dados, porém ao mesmo tempo estas informações apresentam formatos estruturados, semiestruturados e também não estruturados, esses desafios agravam devido as mais variadas fontes de dados no ambiente digital. O conceito de Big Data aplica-se às informações que não podem ser processadas ou analisadas de maneira tradicional utilizando processos e ferramentas convencionais [3][4][5][6].

O tema em questão apresenta algumas propriedades relevantes de modo a diferenciar as tecnologias tradicionais. Alguns autores descrevem apenas três propriedades básicas, são elas, 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 estruturado, semiestruturado e não estruturado), Volume (corresponde as suas tecnologias que superam a armazenagem de dados convencionais, ultrapassando os terabytes e pentabytes), e Velocidade (neste contexto mede o tempo de criação e agregação dos dados. A análise deve ser realizada em microssegundos, decisões devem ser tomadas a respeito dos dados capturados para possuir relevância quando combinados com outros dados) [4][5][6].

As propriedades apresentadas refletem a grandiosidade desse conceito e suas tecnologias que asseguram estas propriedades, devido ao aumento nos fluxos de informações sendo criadas, transmitidas, consultadas e analisadas [7][5].

Os desafios causados pelo aumento de dados estimulam as pesquisas e trabalhos para que soluções confiáveis sejam utilizadas em campos acadêmicos ou organizacionais, oferecendo enumeras formas de geração e reaproveitamento de dados. Existem variedades de métodos, aplicações e ferramentas desenvolvidas para processar grandes volumes de dados [4]. Porém as tecnologias tradicionais são eficientes para trabalharem somente com dados estruturados, e como a grande maioria dos mesmos são gerados 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 cenário.

Com as adversidades causadas por problemas relacionados as tecnologias tradicionais, uma solução para enfrentar os desafios no contexto Big Data em relação ao volume, variedade e velocidade 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.

NoSQL não define apenas 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 esquemas ou esquemas flexíveis, suporte à replicações nativas e acesso via APIs 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 [2][8].

III. EXTRAÇÃO DE DADOS

A extração de dados tem como objetivo fornecer maneiras de obter dados de diferentes fontes heterogêneas, como da Web, de modo a oferecer informações para uma base de dados ou algum outro aplicativo. A internet hoje possui diversas informações disponíveis em documentos HTML, que podem ser de grande utilidade para empresas, assim a extração e recuperação de dados com base em suas técnicas e tecnologias visam a captura de tais informações.

A partir da utilização de software de extração de dados para a localizar, coletar e organizar dados de interesse apresentados em diversos formatos, é possível o enriquecimento de grandes bases de dados, permitindo a realização de consultas e cruzamento dos mesmos, que não eram possíveis através das interfaces de consulta pré-estabelecidas das fontes de informação, o qual oferece uma maior integridade [9][10].

No cenário apresentado (a Internet) existem diversos meios de representação de dados, cada domínio é único, por possuir sua própria forma de estruturar as informações que serão disponibilizadas, tornando-se assim um grande desafio para qualquer sistema de extração de dados, sendo de suma importância a análise da fonte que pretende-se extrair. Diversas páginas são compostas por barreiras que impedem a extração na requisição, sendo necessário o envio de parâmetros essenciais para a identificação de que um ser humano está realizando esta tarefa. As barreiras que tais páginas podem possuir impedindo que softwares executem métodos extração são, o uso de cookies ou JavaScript, Captchas, bloqueio de IP da rede do requisitante e mudanças nas continuas nas estruturas das páginas [11].

Existem inúmeras técnicas e tecnologias para extrair informação a partir da análise de páginas, entre estas está o Data Scraping um software capaz de extrair dados da saída de um outro programa, esse modelo é mais conhecido popularmente nos dias atuais como Web Scraping - principal produto deste trabalho - um software capaz de extrair dados de documentos Web, com base em estruturas HTML ou XML, possibilitando a manipulação de dados contidos no corpo (body) das páginas [12][11]. A extração de dados utilizando este software é realizada através da simulação da navegação de um ser humano, que podem incluir a comparação on-line de preços, monitoramento de dados, detecção de mudanças em Websites além de pesquisas e integração de dados [13].

Um scraper, contém um componente a mais de inteligência,

que permite localizar informações em saídas muitas vezes não estruturadas, como é o caso de saídas de programas destinados a serem vistas por seres humanos. Para este projeto será utilizado a arquitetura de um scraper, por possuir a características necessárias para seu desenvolvimento, de tal modo a oferecer as informações requisitadas pelos usuários.

IV. METODOLOGIA

A grande massa de dados disponibilizados em diversos domínios permite a especialistas de dados usufruírem dessas oportunidades. Para isso este projeto concentra na criação de um serviço baseado no estilo de arquitetura RESTful, com a principal função de extrair dados contidos na Internet por meio da parametrização de um agente extrator de dados.

O estilo de arquitetura RESTfull é bastante utilizado hoje por possuir os quatro verbos essenciais que o especificam, são eles GET, POST, PUT e DELETE, e por ser utilizado para a transferência de dados em formatos semiestruturados como XML ou JSON [14]. Neste projeto a construção da API será baseada neste modelo, permitindo sua utilização posterior, por qualquer sistema que atendam suas especificações necessárias.

A API possuirá recursos que garantam a confiabilidade e segurança das transferências das informações. Porém seu maior enfoque será na parametrização de métodos para extração de dados, em diversos domínios da Web, a fim de extrair informações no formato Big Data, com a disponibilização deste serviço.

A parametrização se dá por meio do usuário, em que o mesmo criará seus próprios robôs de busca, com a transferência de elementos necessários para a realização da extração de informações de seu interesse.

Para o estabelecimento das técnicas que serão utilizadas para a 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, entre qualquer outro atributo presente no corpo da página (body), além de informar parâmetros que remetem a restrições dos domínios como por exemplo, formulários de consultas, sendo evidente o envio de parâmetros como, métodos de requisição e campos presentes a ele, e por fim o formato de retorno dos dados que serão extraídos, no caso o retorno é um JSON.

A Figura 1 apresenta toda a arquitetura deste serviço, como forma de proporcionar a compreensão do mesmo. Neste trabalho os esforços de desenvolvimento estão direcionados para a consolidação de diversos métodos de extração de dados, presentes na API e concretização de seu uso por meio do serviço Web.

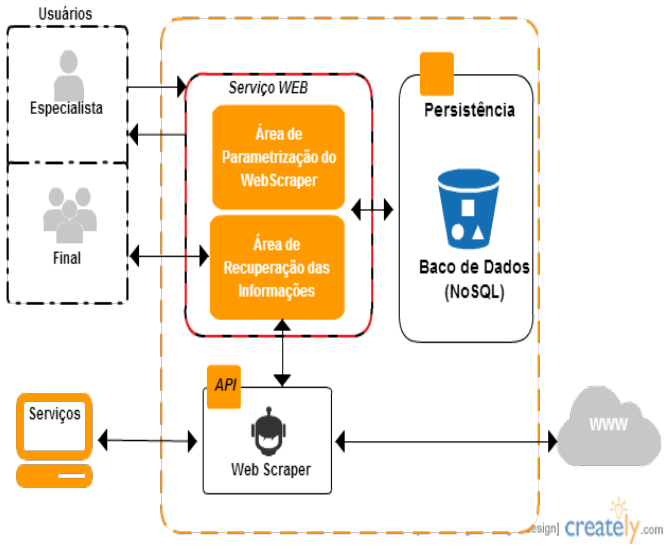

Figura 1 - Representação da Arquitetura de serviços.

Para o desenvolvimento do scraper parametrizado, será essencial o uso de ferramentas e bibliotecas que permitam a navegação entre as páginas HTML e a sua leitura, além do entendimento sobre as atividades que o mesmo exerce na Web. Com base nas pesquisas realizadas a linguagem que permite uma maior rapidez e eficiência no estabelecimento das regras para a extração de dados é o Ruby e seu framework Rails, proporcionando um maior conjunto de métodos para este desenvolvimento.

Como exemplificado no decorrer do texto e na Figura 1, 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.

A. 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 2, 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 1.

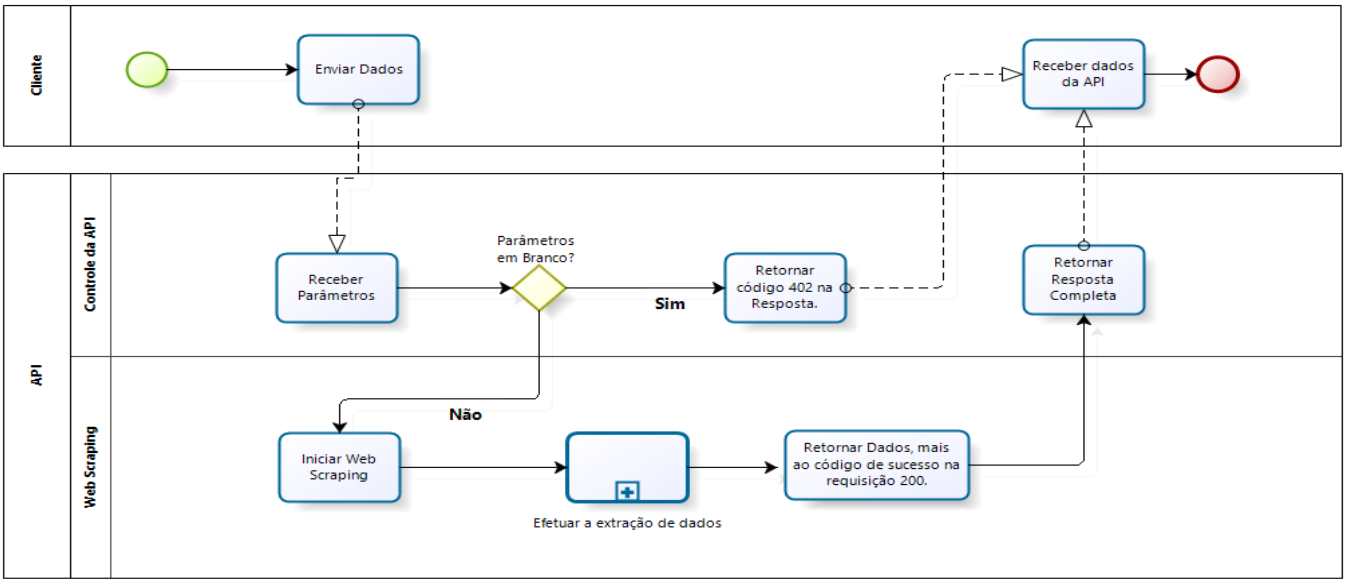

Figura 2 - Representação da Funcionalidade da API.

Tabela 1 - Parâmetros de consulta.

Nº	Padrão URI	Métodos	Operação	Parâmetros de Entrada
1	/scrapers	GET	Consultar	URL, {h1, table} e XML ou JSON.
2	JSON	GET	Consultar	

Utilizando 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 2. 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 presentes 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 2 - Descrição dos códigos presente na API.

Código	Mensagem	Descrição de 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	The data does not exist in the source site's database.	**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.

1) Atividade de um Web Scraper.

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

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

```
1 $ -> gem install mechanize
```

Figura 3 - Representação do download da gem Mechanize.

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. 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 4, 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.

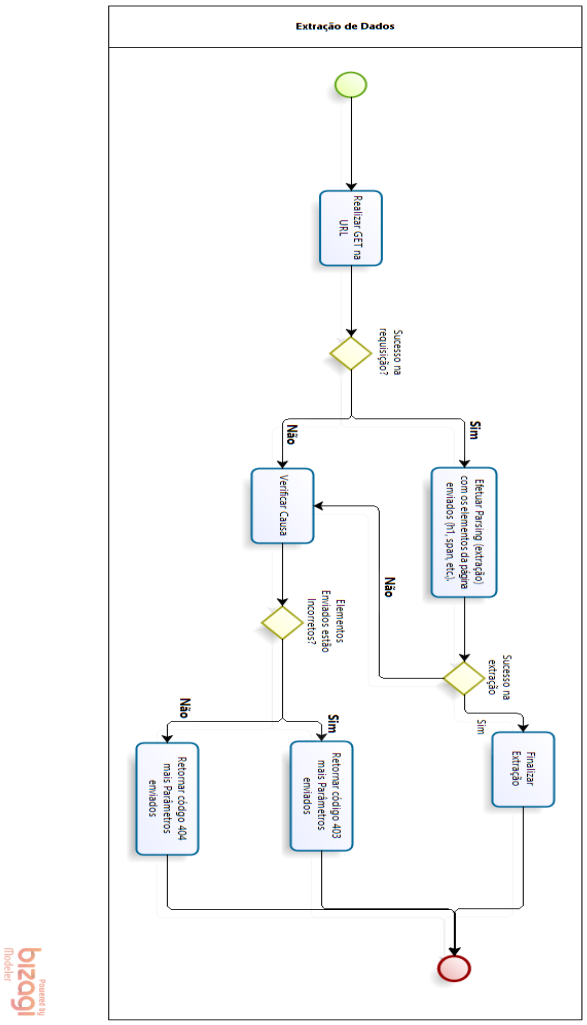

Figura 4 - Representação da funcionalidade do Web Scraping.

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.

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 [14]. 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.

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.

B. Especificações do Serviço Web Cliente

O serviço tem por objetivo demonstrar 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 Scraping, “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.

V. 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.

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.

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.

REFERÊNCIAS

[1] M. DEITEL, J. DEITEL, R. NIETO. “Internet & World Wide Web Como Programar”. 2 ed.: Bookman, 2003.

[2] 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.

[3] ZIKOPOULOS, Paul C., et al. “Understanding Big Data: Analytics for Enterprise Class Hadoop and Streaming Data”. 1.ed. New York, NY. McGraw-Hill, 2011.W.-K. Chen, *Linear Networks and Systems*. Belmont, CA: Wadsworth, 1993, pp. 123–135.

[4] SAGIROGLU, Seref e SINANC, Duygu. “Big data: A review”. Em: Collaboration Technologies and Systems (CTS), 2013 Conferencia Internacional. em IEEE, 2013. p. 42-47.

[5] KAKHANI, K., KAKHANI, S., e BIRADAR. “Research Issues in Big Data Analytics”. vol 2. Agosto 2013. Disponível em: http://www.ijaiem.org/volume2issue8/IJAIEM-2013-08-29-070.pdf. Acessado em: abril de 2014.

[6] KAKHANI, K., KAKHANI, S., e BIRADAR. “Research Issues in Big Data Analytics”. vol 2. Agosto 2013. Disponível em: http://www.ijaiem.org/volume2issue8/IJAIEM-2013-08-29-070.pdf. Acessado em: abril de 2014.

[7] KATAL, WAZID e GOUDAR. “Big Data: Issues, Challenges, Tools and Good Practices”. Disponivel 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.

[8] DIANE, GEROSA. “Utilização da Classe de Banco de Dados NOSQL como Solução para Manipulação de Diversas Estruturas de Dados”. Disponivel em: http://ulbra-to.br/encoinfo/artigos/2012/Utilizacao_da_Classe_de_Banco_de_%20Dados_NOSQL_como_Solucao_para_Manipulacao_de_Diversas_Estruturas_de_Dados.pdf. Acessado em: maio de 2014.

[9] BAUMGARTNER, et. al. “Web Data Extraction for Business Intelligence: the Lixto Approach”. Disponivel em:http://pdf.aminer.org/000/069/407/web_data_extraction_for_business_intelligence_the_lixto_approach.pdf. Acessado em: março de 2014.

[10] LAENDER, H. F. et. al. “A brief survey of web data extraction tools”. Disponivel em: http://www.sigmod.org/publications/sigmod-record/0206/laender-survey.pdf. Acessado em: março 2014.

[11] IMPERVA. “Detecting and Blocking Site Scraping Attacks”. Disponivel em: http://www.imperva.com/docs/wp_detecting_and_blocking_site_scraping_attacks.pdf. Acessado em: março de 2014..

[12] O'REILLY, Radar Team. “Big Data Now: Current Perspectives from O'Reilly Radar”. Eds: “O'Reilly Media, Inc.”, 2011.

[13] CATANESE, MEO, FERRARA, FIUMARA e PROVETTI. “Crawling Facebook for Social Network Analysis Purposes”. Disponivel em: http://arxiv.org/pdf/1105.6307.pdf. Acessado em: abril de 2014.

[14] RICHARDSON, Leonard; RUBY, Sam. RESTful serviços web. Rio de Janeiro: Alta Books, 2007.

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