Web Crawler - Explorer

Carlos Henrique P. da Silva
Lucas Gabriel de Moraes Martins

Bacharelado em Ciência da Computação
Centro Universitário SENAC - Campus Santo Amaro (SENAC-SP)
Av. Engenheiro Eusébio Stevaux, 823 – Santo Amaro, São Paulo – CEP 04696-000 – SP – Brasil
lucas.gmmartins@gmail.com e carloshpds@gmail.com

Resumo

Este pôster tem como assunto o desenvolvimento de um web crawler, também chamados de indexadores ou bots, e utilizados para percorrer a web a procura de realizar extração de dados e/ou meta-dados de conteúdos tanto estáticos quanto dinâmicos de páginas web.

1. Introdução

WEb Crawler, considerado um agente de software, tem como função navegar na WWW (*World Wide Web*),rastreando informações, dados ou meta-dados nas páginas web.

2. Objetivos

O Crawler desenvolvido possui um objetivo de entendimento simples, trazer meta-dados (tags HTML) com suas respectivas referências, chamadas de *hypertext references* ou de *sources*, entre os mesta-dados escolhidos para a visualização estão:

- Links;
- Vídeo;
- Imagens;
- Documentos.

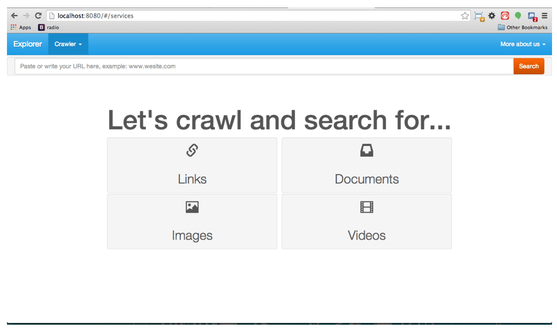
3. Metodologia

Na criação do Web Crawler foi realizado com a poderosa e expressiva linguagem Python em sua versão 2.7 e com o sistema de gerenciamente de banco de dados SQLite, também foi utilizado no projeto o framework CherryPy como servidor web com o objetivo de fornecer objetos estáticos como HTML e CSS para o client. Para o parse do HTML foi usado o BeautifulSoup. Uma das técnicas foi utilizar SSE (Server Sent Events), com fins de tornar a comunicação de server-client rápida e por sua vez a visualização de dados.

O Python é uma linguagem de programação que possue estruturas de dados de alto nível, a linguagem se torna ideal para desenvolvimento rápido de aplicações de diversas áreas e na maioria das plataformas por ser de fácil aprendizado e totalmente expressiva.

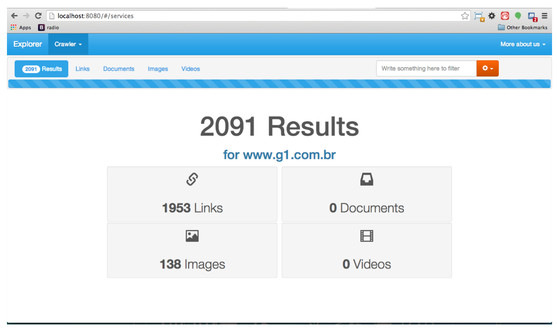
Para a realização do front-end foi utilizado a linguagem CoffeeScript/JavaScript, estilização de páginas com SASS/CSS/CSS3, markup com HTML/HTML5, com a utilização Yeoman para scaffolding, modern workflow e gerenciamento de pacotes juntamente com o Bower, além da incorporação de frameworks como: AngularJS, UnderscoreJS, GruntJS, Bootstrap (Twitter), Compass e JQuery.
O *workflow* da aplicação é baseado em cinco passos, sendo:

- Receber o site onde se deseja efetuar a busca dos meta-dados;
- Disparar o crawler no site desejado;
- Armazenar em lotes os meta-dados encontrados pelo crawler no banco de dados;
- Recuperar os items salvos no banco de dados;
- Repassar os items para o *client.*

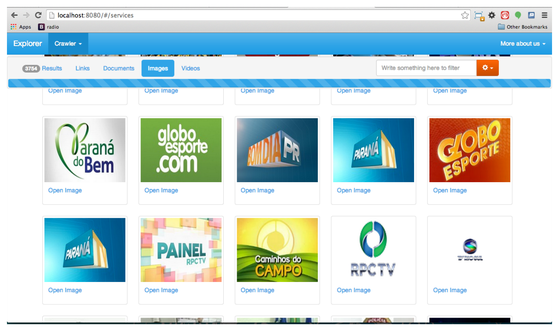


4. Resultados e Discussão

Após várias versões de arquiteturas para a o crawler, encontrou-se uma qual realiza o objetivo para qual foi desnvolvido com êxito e que responda rapidamente aquilo que se achava no website em questão.



Assim, buscas em sites ou portais de grande porte podem ser indexados em tempo real e o usuário não precisa esperar muito tempo para ter algo palpável a sua frente, pois consegue obter resultados por demanda.



5. Conclusão

A extração dos meta-dados foi aplicada com êxito, de maneira funcional, porém sugestiva a mudanças para melhoria de precisão na coleta do tipo de dado do item em questão.

Referências

[1] Menezes, Nilo **Python Software Foundation, DocumentacaoPython**. *Disponível em:* http://www.python.org.br/wiki/DocumentacaoPython, Acesso em: 20 de out. 2013.