UNIVERSIDADE FEDERAL DE SANTA CATARINA

Crawler Webservice para auxiliar no monitoramento de bolsas e recursos para Instituições de Ensino Superior

Brian Henkels

Florianópolis - SC

2012/2

UNIVERSIDADE FEDERAL DE SANTA CATARINA

DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA

CURSO DE SISTEMAS DE INFORMAÇÃO

Crawler Webservice para auxíliar no monitoramento de bolsas e recursos para Instituições de Ensino Superior.

Brian Henkels

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Florianópolis - SC

2012/2

Brian Henkels

Crawler Webservice para auxíliar no monitoramento de bolsas e recursos para Instituições de Ensino Superior.

Trabalho de conclusão de curso apresentado como parte dos requisitos para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: **Prof. Dr. João Cândido Dovicchi**

Banca Examinadora

Profa. Dra. Carina Friedrich Dorneles

Prof. Dr. Mário Antônio Ribeiro Dantas

Dedicatória

Dedico este trabalho aos meus pais, meus irmãos e minha sobrinha e aos amigos que em nenhum momento deixaram de me apoiar e incentivar, apesar de alguns problemas enfrentados.

Agradecimentos

Agradeço a todos que me auxiliaram na realização deste trabalho, familiares, amigos, professores, colegas de trabalho e colegas da graduação.

Sumário

Lista de ilustrações

Lista de tabelas

Lista de Reduções

UFSC – Universidade Federal de Santa Catarina

INE – Departamento de Informática e Estatística

WWW – World Wide Web

URL – *Uniform Resource Locator* - Endereço da web

PHP – um acrônimo recursivo para "PHP: Hypertext Preprocessor", originalmente Personal Home Page

Cnpq – Conselho Nacional de Desenvolvimento Científico e Tecnológico

Setic – Superintendência de Governança Eletrônica e Tecnologia da Informação e Comunicação.

HTML – HyperText Markup Language – Linguagem de Marcação de Hipertexto

CGI – Common Gateway Interface

XML – eXtensible Markup Language

XHTML – eXtensible Hypertext Markup Language

PDF – Portable Document Format

Doc – Arquivo documento

ODBC – Open Data Base Connectivity – Padrão Aberto de Conexão com Bancos de Dados

POSIX – Portable Operating System Interface – Portável entre Sistemas Operacionais

BSD – Berkeley Software Distribution – Sistema Operacional UNIX desenvolvido pela Universidade de Berkeley, na Califórnia

MVC – Model-view-controller – Um padrão de projeto

GPL – General Public License – Licença Pública Geral

SQL – Structured Query Language – Linguagem de Consulta Estruturada

OLAP – On-line Analytical Processing

GIS – Geographic Information System

SOAP - Simple Object Access Protocol – Protocolo Simples de Acesso a Objetos

W3C – The World Wide Web Consortium – Consórcio World Wide Web

WSDL – Web Services Description Language – Linguagem de Definição de Web Services

LISA - Laboratory for Integration of Information Systems and Advanced Applications – Laboratório para Integração de Sistemas de Informação e Aplicações Avançadas

URL – Uniform Resource Locator – Localizador Padrão de Recursos

Resumo

No Brasil hoje existem incontáveis instituições de ensino, públicas ou privadas, que recebem incentivos do governo e de empresas dedicadas à área de pesquisa. Estes incentivos normalmente vêm por meio de bolsas e recursos para que possam subsidiar pesquisas realizadas pelos professores e por alunos, tanto da graduação, quanto pós-graduação, mestrado e doutorado.

Em um ambiente de milhares de possíveis beneficiários, apenas na UFSC, controlar este tipo de informação se torna extremamente trabalhoso e demanda tempo.

Para a solução deste problema é proposto o desenvolvimento de um sistema, como um *webservice,* utilizando um *Crawler,* que vasculhe os sites para encontrar possíveis informações sobre os recursos, aprovados ou não, e as retorne de forma organizada e como uma ordem de importância previamente avaliada.

Palavras-chave: *Crawler*, Robô, PHP, Zend Framework, MySQL, Linguagem Natural, *Full Text*, *Stop Words*, *WebService*, SOAP, WSDL, Expressão Regular, XPDF

Abstract

In Brazil there are now countless educational institutions, public or private, that receive incentives from the government and companies dedicated to the research area. These incentives usually come through scholarships and resources that can support research conducted by teachers and students, both graduate, and post-graduate, masters and doctorate.

In an environment of thousands of potential beneficiaries only at UFSC, control this type of information is extremely laborious and time consuming.

To solve this problem we propose the development of a system as a webservice using a crawler that scour the websites to find information on possible resources, approved or not, and return in an organized manner and as a pre-order of importance evaluated.

Keywords: Crawler, Robot, PHP, Zend Framework, MySQL, Natural Language, Full Text, Stop Words, WebService, SOAP, WSDL, Regular Expression, XPDF

Introdução

A Universidade atualmente conta com uma comunidade de algumas dezenas de milhares de alunos e professores que, em grande quantidade, provêm conhecimento para a sociedade por meio de programas de pesquisa, bolsas de iniciação científica, entre outras. Devido a isso, a UFSC hoje está incluída na lista das universidades ao redor do mundo que possuem uma grande produção acadêmica e nível intelectual elevado de acordo com dados do Ministério da Educação.

As pesquisas das universidades normalmente são subsidiadas por instituições de apoio como FINEP, CAPES, CNPq etc.. Assim, o volume de informação que precisa ser controlado diariamente cresce e a instituição não consegue se manter atualizada sobre resultados de recursos aprovados, o que pode causar atrasos para o início de projetos.

Da necessidade de automatizar o processo surgiu a idéia de desenvolver uma aplicação que busque as informações nas páginas e nos arquivos disponibilizados nos portais, listando apenas informações relevantes, relacionadas às palavras solicitadas. As informações devem ser retornadas de forma organizada, que seja possível identificar facilmente qual a importância do conteúdo encontrado pelo usuário.

Uma forma de se implementar uma aplicação que possibilite estes recursos é por meio de um *Crawler*, que é um programa que navega por muitas páginas, neste caso, em páginas específicas que contenham disponibilizadas as informações, e indexando a informação encontrada de forma a poder listar dados que tenham uma certa importância. E para simplificar, torná-lo modular e disponibilizá-lo como um

Webservice para ser utilizado por um sistema externo, desenvolvido pela equipe do Setic da UFSC.

Problema

O problema consiste na necessidade de identificar as páginas que contenham conteúdo importante para a universidade, dentro do contexto dos portais de organizações que fornecem recursos de pesquisa como FINEP, CAPES, CNPq, entre outras, como recursos aprovados, informações de professores e outros assuntos.

O *Crawler* precisa ser capaz de identificar de forma genérica, apenas relacionando o conteúdo indexado à informação que o usuário deseja.

As informações são divulgadas diariamente nos portais das instituições, o que faz com que precisem ser consultados na mesma frequência. Muitas vezes, devido à grande quantidade de informação nos portais, a tarefa de encontrar a informação desejada se torna trabalhosa. Na primeira pesquisa sobre estas informações o sistema deve manter o histórico desta consulta e passar a atualizá-la diariamente para que a próxima vez que esta busca for feita, sua resposta seja retornada em questão de segundos, quando levariam severos minutos para retornar caso fossem feitas na hora.

Objetivos

Geral

Criar um processo e desenvolver uma aplicação capaz de buscar informações acadêmicas de interesse da universidade, um protótipo de software utilizando *WebService* que retorne os dados de forma organizada, por intermédio de um *Crawler*.

Específicos

- Desenvolver uma aplicação que possa ser utilizada por outras aplicações, independente da plataforma – portabilidade;
- Fazer o levantamento de referências;
- Pesquisar o estado da arte em *webservices* + *crawlers*;
- Determinar uma forma de buscar informações textuais em portais com grande quantidade de páginas que podem ser texto estático, formulários ou mesmo arquivos;
- Organizar os textos de forma a poder encontrar informações relevantes em relação a um ou mais termos procurados.

Este trabalho esta organizado em:

- Introdução – Breves comentários sobre o trabalho assim como do problema a que este trabalho se propõe a solucionar.
- Referencial Teórico – Descrição e especificação dos conteúdos necessários para o desenvolvimento da aplicação.
- Materiais e Métodos – Técnicas e materiais que foram necessários e utilizados durante o desenvolvimento.
- Resultados – Análise de alguns testes realisados para provar a eficácia e eficiência do sistema.
- Conclusão

Referencial Teórico

Antes de apresentar as técnicas de desenvolvimento e as formas de pensamento do projeto serão apresentadas as tecnologias necessárias para o decorrer do trabalho.

Crawlers

Crawler ou *Web Crawler* (rastreador em inglês), também conhecido como *robot (*robô), *spider* (aranha) e alguns outros nomes, é um programa que vasculha uma ou mais regiões da *World Wide Web* de forma automatizada por informações.

Os *Crawlers* são quase tão antigos quanto a própria *web* e alguns artigos foram apresentados nas primeiras conferências sobre a internet realizadas no início dos anos 90. Entretanto, naquela época, o ambiente da web era muitas vezes menor do que é nos dias de hoje e isso se tornou um problema para os *crawlers.* (HEYDON NAJORK, 1999)

Todos os mecanismos de busca populares utilizam *crawlers* que vasculham a web 24 horas por dia para indexar as informações, mas devido à competição entre os mecanismos de busca, a arquitetura dos crawlers não é públicamente descrita, ou são tão complexos que os tornam praticamente impossível de se reproduzir, como por exemplo, o Google que desenvolveu um crawler distribuído utilizando múltiplas máquinas.

O *crawler* do Google vasculha regularmente a *web* para reconstruir a indexação. A busca é baseada em muitos fatores como *PageRank,* links para uma página e regras de busca como o número de parâmetros de uma URL.

WebServices

Fruto da disseminação da computação distribuída, os *WebServices* surgiram para integrar e conectar diferentes informações externas e internas, estejam os dados em servidores, em estações de trabalho ou até mesmo em mainframes. Em resumo, a tecnologia permite que dispositivos conectados à Internet troquem mensagens entre si, sem a intervenção direta dos usuários.

Um *WebService* é um componente que possui suas funcionalidades acessíveis pela rede através de troca de mensagens baseadas em XML (*eXtensible Markup Language).* A disponibilização das operações e a descrição do serviço também ocorrem através do padrão XML. O arquivo descritor do serviço possui todas as informações necessárias para que outros componentes possam interagir com o serviço, incluindo o formato das mensagens para as chamadas aos métodos do serviço, protocolos de comunicação e as formas de localização do serviço. Um dos maiores benefícios dessa interface é a abstração dos detalhes de implementação do serviço, permitindo que seja acessado independente da plataforma de hardware ou software na qual foi implementado.

Como as mensagens trocadas para a comunicação são baseadas no padrão XML, também tem-se a flexibilidade com relação à linguagem de programação tanto na implementação do serviço quanto no componente que acessará o *WebService*. Estas características permitem e motivam a implementação de aplicações Web baseadas em *WebService* por torná-las fracamente acopladas com as outras partes do código da aplicação. Com isso, as aplicações adquirem uma arquitetura orientada a componentes e tornam-se flexíveis com relação às várias plataformas disponíveis no mercado. Um *WebService* geralmente é implementado para disponibilizar uma determinada

funcionalidade visando a reusabilidade do WebService e a interoperabilidade com outros sistemas.

Uma grande vantagem dos *WebServices* reside no fato de a equipe de desenvolvimento poder focar seus esforços no sistema em si, praticamente sem se preocupar com o meio de comunicação entre os processos. Especialmente para as grandes corporações – que possuem uma infinidade de soluções concebidas por fornecedores diferentes ou mesmo desenvolvidas internamente nas mais variadas plataformas – esse conceito promete unificar, pela *Web*, todas as informações contidas nas aplicações, sem que haja necessidade de migração.

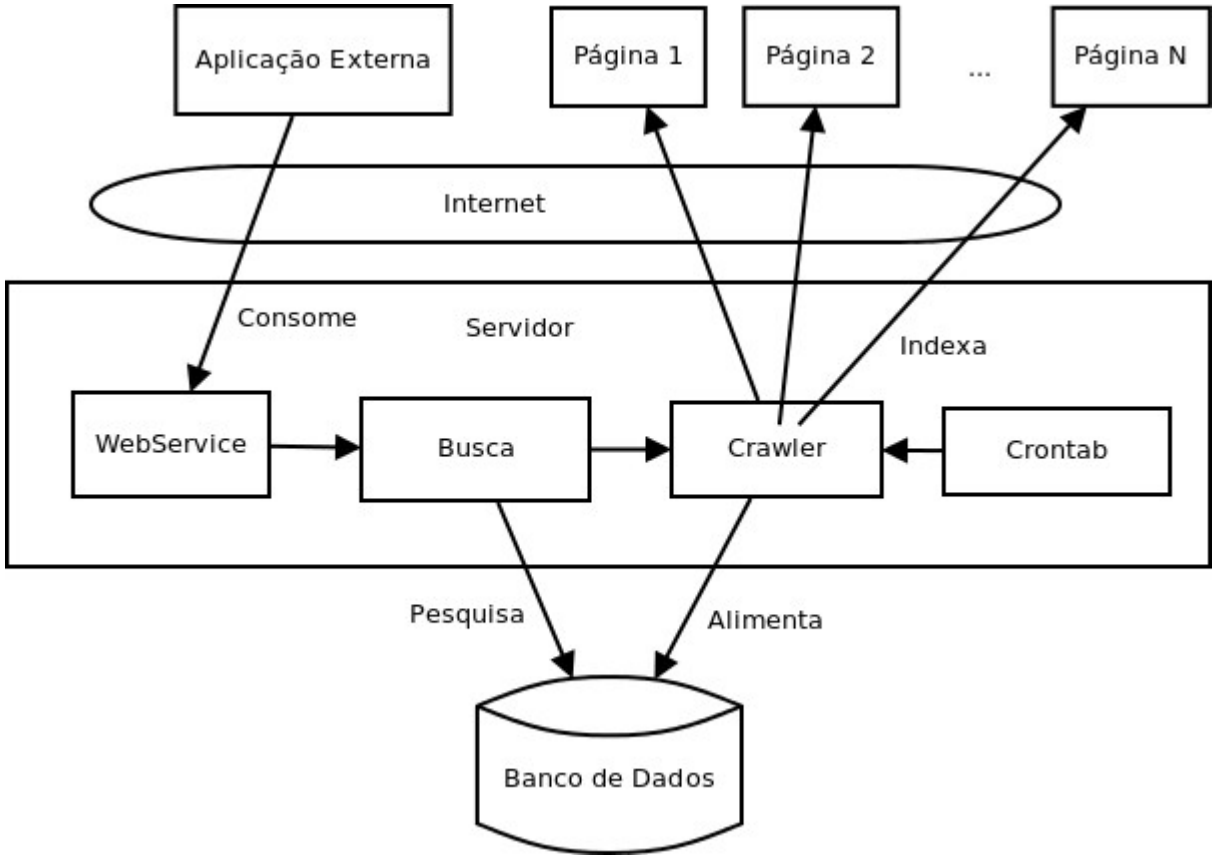

Ilustração 1: Proposta de Componentes e Comunicação

Materiais e Métodos

Tecnologias Utilizadas

Linguagem PHP com Zend Framework

Para o desenvolvimento da aplicação, é utilizada a linguagem de programação PHP, que atualmente é a linguagem mais usada no mundo para aplicações direcionadas para a web. A aplicação se baseia no framework Zend, facilitando o desenvolvimento de algumas funcionalidades.

A linguagem de programação PHP é amplamente utilizada em todos os tipos de aplicações Web e pode ser aplicada diretamente no HTML. O PHP é focado para ser uma linguagem de script do lado do servidor, portanto, pode fazer qualquer coisa que outro programa CGI pode fazer, como coletar dados de formulários, gerar páginas com conteúdo dinâmico ou enviar e receber cookies.

O script no lado do servidor (server-side) é o mais tradicional e principal campo de atuação do PHP. Para ser utilizado são necessários três componentes: o interpretador PHP (como CGI), um servidor e um navegador. Ele também pode ser utilizado diretamente através de linhas de comando sem necessitar de um servidor web e um navegador, necessitando apenas do interpretador, sendo ideal para a execução de rotinas usando o cron. Existe também o PHP-GTK que é uma extensão do PHP, não disponibilizada na distribuição oficial. O PHP-GTK permite escrever aplicações desktop com interface gráfica multi-plataforma, apesar de que possivelmente não é melhor linguagem para criação deste tipo de aplicação.(PHP.net, 2013)

O PHP pode ser utilizado na maioria dos sistemas operacionais, incluindo *Linux*, várias variantes *Unix, Microsoft Windows, Mac OS X, RISC OS,* e outros. O PHP também é suportado pela maioria dos servidores web atuais. (PHP.net, 2013)

O PHP, portanto, permite escolher o sistema operacional e o servidor web, assim como entre utilizar programação estrutural ou orientada a objetos, ou ainda uma mistura dos dois. A partir da versão 5 do PHP, grande parte dos recursos da programação orientada a objetos foram implementados.

Com o PHP não se restringe apenas a gerar HTML, permitindo a geração de qualquer padrão de texto como XML e XHTML, de imagens, documentos (PDF, Doc, etc) e até animações *flash* criados dinamicamente.

Uma das características mais importantes do PHP é o seu suporte a uma ampla variedade de bancos de dados e o fato de estes recursos serem muito simples de se utilizar. O PHP suporta ODBC (*Open Database Connection*, ou Padrão Aberto de Conexão com Bancos de Dados), o que permite que se utilize qualquer banco de dados que suporte este padrão mundial.(PHP.net, 2013)

Um dos pontos mais importantes para este trabalho é que o PHP é extremamente útil em recursos de processamento de texto, tanto como interpretador para documentos XML ou expressões regulares e o POSIX Estendido.

O Zend Framework é um framework baseado em PHP, totalmente orientado a objetos, utilizando o paradigma MVC e licenciado como *New BSD License*, possuindo uma ampla gama de recursos para auxiliar no desenvolvimento. Possui contribuidores de software livre e que assumem responsabilidade pelo fato de seu código não ser de propriedade intelectual de terceiros. Este framework fornece código limpo, estável, completo e com direitos de propriedade intelectual livre.

Mysql Community Edition

O banco de dados a ser utilizado, escolhido para esta aplicação foi o MySQL, por ser um banco de dados de código aberto, com ampla documentação e funcionalidades importantes para o projeto. A facilidade de utilização deste pela linguagem PHP e a técnica de indexação *Full Text* são as principais características que levaram à esta escolha.

O MySQL é o sistema gerenciador de bancos de dados de código aberto mais popular do mundo e está disponível sob a licensa GPL, utiliza a linguagem SQL (Linguagem de Consulta Estruturada, do inglês *Structured Query Language*) como interface, além de possuir uma extensa e ativa comunidade de desenvolvedores a mantendo. (MySQL.com, 2013)

O MySQL hoje suporta *Unicode*, *Full Text Indexes*, replicação, *Hot Backup*, *GIS*, OLAP e muitos outros recursos de banco de dados.inclui algumas funcionalidades importantes nos bancos de dados como replicação, particionamento, stored procedures, triggers (gatilhos), visões, entre outras e está disponível para mais de vinte plataformas que incluem Linux, Unix, Mac e Windows. (MySQL.com, 2013)

SOAP

A arquitetura cliente-servidor webservice a ser utilizada é suportada por algumas técnologias e uma das mais utilizadas é o SOAP que permite aplicações terceiras acessarem as funcionalidades fornecidas. São definidas regras para enviar e receber chamadas remotas, assim como a estrutura dos dados enviados e recebidos pelas funcões da aplicação.

O SOAP é um dos principais componentes da tecnologia *WebServices*. Surgiu no ano de 1998, apresentado ao *World Wide Web* Consortium pelas empresas

DevelopMentor, Microsoft e UserLand Software como um Internet Draft. Inicialmente, este protocolo definia um mecanismo para transmissão de procedimentos remotos utilizando XML sobre HTML. Devido a divergências políticas, sua especificação em prevista para 1998 não ocorreu, mas sim em dezembro de 1999.

Um dos grandes benefícios do SOAP é que ele é aberto e foi adotado pela maioria das grandes empresas de hardware e software. A sua especificação provê a base para a comunicação aplicação-aplicação: os WebServices. Construído no topo de padrões abertos como HTTP e XML, facilita o aprendizado, por parte dos desenvolvedores e o suporte das infra-estruturas. (BRAGA HENRIQUE BORGES GOMES, 2011)

WSDL (WebServices Description Language)

Na implementação de *webservices* é possível disponibilizar uma documentação que descreve todas as funcionalidades disponíveis para facilitar a utilização por terceiros. Esta documentação é encontrada no endrereço *web* que o webservice estiver disponível. Será utilizado este recurso na aplicação e aqui será dada uma breve explicação sobre isso.

WSDL define um sistema para a descrição de serviços. Através dela, descrevem-se os serviços externos, ou interfaces que são oferecidas por uma determinada aplicação, independente de sua plataforma ou linguagem de programação.

Seu principal objetivo é descrever as interfaces apresentadas e apontar a localização dos seus serviços, disponíveis em um local previsível e bem conhecido, na rede, o qual permite que o cliente acesse de maneira confiável. Por ser um documento XML, sua leitura se torna fácil e acessível.

É a linguagem de descrição de WebServices baseada em XML. Ela permite, através da definição de um vocabulário em XML, a possibilidade de descrever serviços e a troca de mensagens. Mais especificamente é responsável por prover as informações necessárias para a invocação do WebService, como sua localização, operações disponíveis e suas assinaturas. (W3C, 2013)

Conversor de PDF - XPDF - pdftohtml

Algumas das informações disponibilizadas nos portais que serão vasculhados são disponibilizadas em formados arquivos específicos distintos dos comumente utilizados como PDF, Doc etc, que não são facilmente lidos de forma automática pois possuem uma formatação interna específica e algumas delas definidas por código proprietário. O XPDF é um recurso de código aberto, que supre com eficiência a necessidade de transcrição dos conteúdos dos arquivos para o mesmo formato fornecido pelas aplicações web, isto é, em texto plano.

O uso do XPDF é por linha de comando via terminal, e seus comandos são muito simples, que podem tambpem ser utilizados via PHP por meio de chamadas ao sistema operacional.

O recurso principal a ser utilizado neste trabalho é a conversão de arquivos PDF, Doc, Xls para texto plano ou HTML, para que seja tratado como todo o resto do conteúdo das páginas. Muitos arquivos podem possuir recursos de segurança habilitados como encriptação e um recurso interessante do XPDF é que ele inclui código de decodificação.

Utilizando o PHP para converter arquivos PDF:

```
# converte o pdf em html - converte o arquivo1.pdf no arquivo2.html
# parâmetro "-s" : generate single document that includes all pages
```

```
# parâmetro "-i" : ignore images
shell_exec( "pdftohtml -s -i arquivo1.pdf arquivo2.html" );
```

Cron e Crontab

O programa Cron, ou Vixie Cron é um programa nativo dos sistemas operacionais baseados em Linux e Unix que é iniciado automáticamente com o sistema e serve para executar rotinas automaticamente no sistema com horário marcado ou a cada período de tempo. A cada minuto este programa é chamado e executa todas as rotinas agendadas para este minuto.

Crontab é um programa utilizado para gerenciar as rotinas utilizadas na Cron.

Técnicas Utilizadas

Expressões Regulares

O uso de expressões regulares é uma forma flexível e reconhecida de identificar cadeias de caracteres como palavras, frases ou padrões de sequência. Elas são escritas numa linguagem formal que é interpretada servindo como um analisador sintático, validando a composição sequencial dos caracteres com a especificação dada.

Uma expressão regular não precisa listar todos os elementos do conjunto, podendo fazer uso de recursos ou operadores quantificadores, condicionais, agrupadores etc.

As expressões regulares surgiram da teoria dos autômatos e da teoria das linguagens formais que fazem parte da teoria da computação. Estas teorias estudam modelos de computação e formas de descrição e classificação de linguagens formais.

Exemplos de expressões regulares:

.* : qualquer conjunto de cadeias desde o conjunto vazio até o conjunto composto por todos os caracteres existentes, sem restrição de número de ocorrências. (conjunto infinito de cadeias)

.+ : qualquer conjunto de cadeias que possua um ou mais caracteres. (conjunto infinito de cadeias)

a? : {“”, “a”} (conjunto finito de cadeias)

x|y : {“x”,”y”} (conjunto finito de cadeias)

(x|y)* : {“”, “x”, “y”, “xx”, “yy”, “xy”, “yx”, “xyx”, … } (conjunto infinito de cadeias)

[a-z]* : qualquer cadeia formada apenas por letras de “a” a “z” exclusivamente minúsculas e de qualquer comprimento. (conjunto infinito de cadeias)

[^a]* : qualquer cadeia que não contenha o caractere “a”. “^” dentro do conjunto denota negação.

a(b|c)d : {“abd”, “acd”}

a(b|c)?d : {“abd”, “acd”}

[a-z]{3} : qualquer conjunto de combinações de “a” a “z” com exatamente 3 caracteres.

[a-z]{1-3} : qualquer conjunto de combinações de “a” a “z” com no mínimo 1 caractere e no máximo 3 caracteres.

Existem ainda alguns outros caracteres como “$” que denota final de cadeia e “^” no início da expressão, que representa o início da cadeia. Estes símbolos normalmente são utilizados para identificar padrões internos a um conjunto de cadeias, como em busca por palavras em um texto.

Indexação Full Text com buscas em linguagem natural

Nos sistemas gerenciadores de bancos de dados atuais existem inúmeras formas diferentes de indexação de informações, a forma mais comum utilizada é a

indexação simples onde é criado um *hash* com a localização das referências para tornar as buscas mais ágeis.

Uma forma de indexação importante para busca de textos é a indexação por *Full Text.* Técnica bastante útil, mas não muito conhecida, foi criada com base em técnicas de processadores lingüisticos de *Corpus,* fazendo uso de estatísticas de vocabulário dos textos como quantidade de ocorrência de palavras, tamanho das palavras, freqüencia de letras, entre outras.

Um problema deste recurso são os chamados falso-positivos que é atribuir como bom resultado um mau resultado devido à ambiguidade na linguagem natural, que é utilizada pela busca por Full Text.

- linguagem natural

Uma forma de incrementar a precisão dos resultados é alterando a lista de *stopwords* padrão utilizada pelo banco de dados para lista da linguagem em que os textos pesquisados estiverem. Normalmente a lista padrão é definida na língua inglesa. *Stopwords* são palavras chave comuns em uma língua, que devem ser desconsideradas ou levar um peso menor na consideração do conjunto de palavras ou frases.

Parte da lista de *stopwords* padrão:

```
a
about
above
after
again
against
all
am
an
and
any
```

are
aren't
as

Tabela 1: Stopwords língua Inglesa

Parte da nova lista de *stopwords* em língua portuguesa:

último
é
acerca
agora
algmas
alguns
ali
ambos
antes
apontar
aquela
aquelas
aquele
aqueles
aqui
atrás
bem
bom

Tabela 2: Stopwords Língua Portuguesa

A indexação *Full Text* também possui outros recursos que não serão explorados neste trabalho que é o modo da Busca Booleana que utiliza parâmetros da lógica para aprimorar as buscas. Exemplos:

- `apple banana`
  encontra linhas que contenha pela menos uma destas palavras.
- `+apple +juice`
  ambas as palavras.
- `+apple macintosh`
  palavra ``apple'', mas avaliada mais alto se também conter ``macintosh''.
- `+apple -macintosh`
  palavra ``apple'' mas não ``macintosh''.
- `+apple +(>turnover <strudel)`
  ``apple'' e ``turnover'', ou ``apple'' e ``strudel'' (em qualquer ordem), mas avalia

``apple pie" melhor que ``apple strudel".

- `apple*`
  ``apple", ``apples", ``applesauce", e ``applet".
- `"some words"`
  ``some words of wisdom", mas não ``some noise words".

Tabela 3: Exemplos de utilização de busca booleana em full text

(http://dev.mysql.com/doc/refman/5.0/en/fulltext-boolean.html)

Materiais

Os recursos utilizados para o desenvolvimento do projeto são:

1. Computador próprio - para desenvolvimento e testes.
2. Servidor do Centro de Informática (ADMREDE) e Estatística (INE - UFSC) - INF -disponibilizado para os alunos - para testes de integração do WebService.
3. Servidor do Grupo de Bancos de Dados - LISA - para a aplicação e o banco de dados.

Desenvolvimento do Sistema

Objetivo do Sistema

O objetivo do WebService é retornar uma lista de endereços da internet que contenham informações relacionadas aos termos da pesquisa, dentro de um tempo determinado de processamento, em ordem de relevância.

WebService

A seguir, a estrutura WSDL gerada automáticamente especificando as funcionalidades de busca do *webservice:*

```
<definitions name="Busca"
targetNamespace="http://www.crawler-ws.com.br/crawler">
  <types>
    <xsd:schema
     targetNamespace="http://www.crawler-ws.com.br/crawler"/>
  </types>
  <portType name="BuscaPort">
    <operation name="busca">
      <documentation>
      Busca os termos mapeados.</documentation>
      <input message="tns:buscaIn"/>
    </operation>
  </portType>
  <binding name="BuscaBinding" type="tns:BuscaPort">
    <soap:binding style="rpc"
     transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="busca">
    <soap:operation
     soapAction="http://www.crawler-ws.com.br/crawler#busca"/>
      <input>
        <soap:body use="encoded"
     encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="http://www.crawler-ws.com.br/crawler"/>
      </input>
    </operation>
  </binding>
  <service name="BuscaService">
    <port name="BuscaPort" binding="tns:BuscaBinding">
      <soap:address location="http://www.crawler-ws.com.br/crawler"/>
    </port>
```

```
  </service>
  <message name="buscaIn">
    <part name="termos" type="xsd:string"/>
    <part name="url" type="xsd:string"/>
    <part name="qtde" type="xsd:int"/>
    <part name="forcarNovaPesquisa" type="xsd:boolean"/>
  </message>
</definitions>
```

Tabela 4: WSDL gerado automaticamente pelo sistema

Estrutura do Banco de Dados

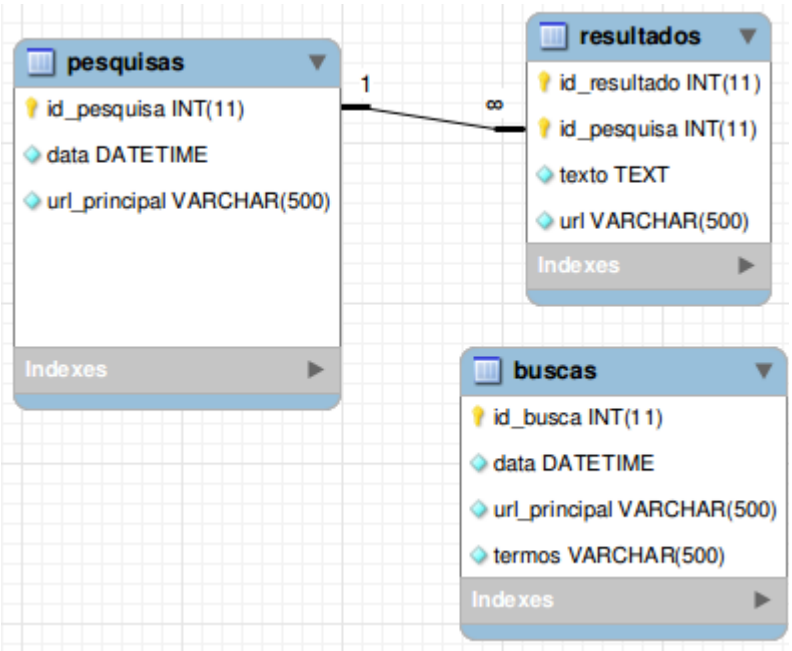

Ilustração 2: Modelo do Banco de Dados

Arquitetura do Sistema

Cliente > WebService > Crawler (processamento do texto) > Banco de Dados (consulta e ordenação) > Crawler > Dados Formatados > WebService > Cliente

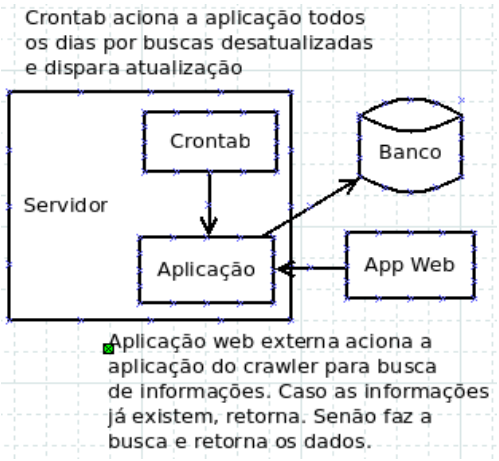

Ilustração 3: Arquitetura do Sistema

Fluxo da Aplicação

1. A aplicação externa consulta o *webservice* fornecendo a *URL* e os termos da busca.

2. Caso não exista nenhuma busca previamente realizada sobre a *URL*, o *Crawler* é acionado para coletar os dados.

3. Retorna o conjunto dos dados, ordenados pelo ranqueamento do índice *full text,* com a *URL*, a frase encontrada (que correspondeu com a busca) e a data em que foi realizada a busca.

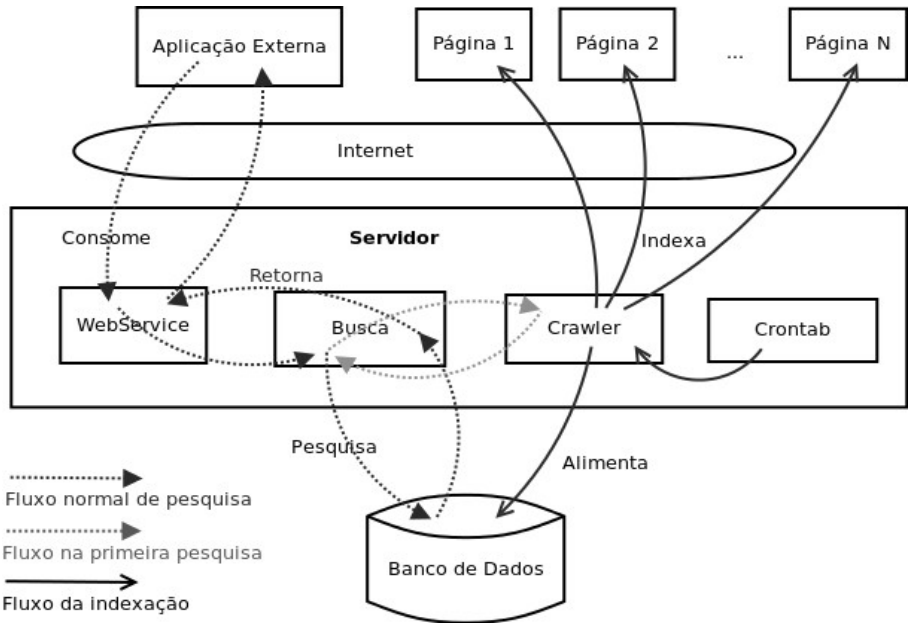

Ilustração 4: Fluxo da aplicação

Crontab – Execução diária

A cada intervalo de tempo (inicialmente definido como 20 minutos), é feita uma consulta que busca entre as *URLs* pesquisadas em um período recente(inicialmente definido como 30 dias), pela mais antiga e atualiza seu resultado. Com isso, as pesquisas que são realizadas frequentemente se mantém sendo atualizadas diariamente, até um limite de 72 pesquisas distintas por dia.

Extração dos textos das páginas no domínio solicitado

O Crawler executa os seguintes passos para extrair as informações das páginas:

1. Armazena o endereço principal sendo pesquisado;
2. Verifica se o endereço atual já foi pesquisado.
3. Verifica se o endereço atual faz parte do domínio pesquisado (do endereço principal).
4. Caso o endereço referencia um arquivo - não *HTML* - tenta converter para *HTML* utilizando o *XPDF*
5. Busca pelas páginas referenciadas na página atual - links -, utilizando expressão regular, e armazena em um vetor.

```
$ereg = '\<a .*href=[\'|"]([^\'|^"]*)[\'|"]\>.*\<\/a\>';
preg_match_all( "/$ereg/", $html, $matches );
```

6. Extrai os textos do HTML utilizando expressão regular.

```
$ereg = "\>[ ]*([^\n|^\r|^<|^\$|^\#]+)[ ]*<";
preg_match_all( "/$ereg/", $html, $matches );
```

7. Armazena os textos no banco de dados
8. Para cada página referenciada, volta para o passo 2.

Busca e ordenação em linguagem natural nos textos encontrados

Exemplo de SQL gerado para três termos: “resultado;doutorado;ufsc”

```
“SELECT group_concat( distinct r.texto separator ', ' ) as textos
     , r.url
     , sum( MATCH (r.texto) AGAINST ('resultado' IN NATURAL LANGUAGE
MODE) ) AS score_t1
     , sum( MATCH (r.texto) AGAINST ('doutorado' IN NATURAL LANGUAGE
MODE) ) AS score_t2
     , sum( MATCH (r.texto) AGAINST ('ufsc' IN NATURAL LANGUAGE
MODE) ) AS score_t3
     , sum(MATCH (r.texto) AGAINST ('resultado' IN NATURAL LANGUAGE
MODE)+MATCH (r.texto) AGAINST ('doutorado' IN NATURAL LANGUAGE MODE)
+MATCH (r.texto) AGAINST ('ufsc' IN NATURAL LANGUAGE MODE) ) as total
FROM resultados r
WHERE 1
and (  MATCH (r.texto) AGAINST ('resultado' IN NATURAL LANGUAGE MODE)
OR  MATCH (r.texto) AGAINST ('doutorado' IN NATURAL LANGUAGE MODE)  OR
          MATCH (r.texto) AGAINST ('ufsc' IN NATURAL LANGUAGE
MODE)  )
and r.id_pesquisa = '5'
group by r.url
order by total desc
limit 0,30 ;”
```

Tabela 5: Exemplo de Consulta Utilizando Full Text

Retorno dos Dados

Para o retorno do *webservice* foi criado um objeto simples contendo a lista de ocorrências encontradas no banco de dados de acordo com a busca realizada conforme o exemplo a seguir:

Busca em “www.finep.gov.br” por “chamadas já disponíveis”.

Resultado:

```
object(Dados)#39 (1) {

  ["lista"]=>
  array(30) {
    [0]=>
```

```
    array(4) {
      ["textos"]=>
      string(482) "Chamadas Públicas, Chamadas encerradas/resultados,
construção e manutenção das 3 (três) presenças online da FINEP, incluindo
versão para dispositivos móveis., - disponível no Portal COMPRASNET, em
Acesso Livre > Consultas > Atas de Pregões, disponível no Portal COMPRASNET,
em Acesso Livre > Consultas > Atas de Pregões, LEILÃO Nº 01/2005 - VENDA DE
AUTOMÓVEIS "
      ["url"]=>
      string(34) "http://www.finep.gov.br/licitacoes"
      ["score_t1"]=>
      string(18) "226.10632991790771"
      ["total"]=>
      string(18) "226.10632991790771"
    }
    [1]=>
    array(4) {
      ["textos"]=>
      string(316) "Chamadas Públicas, Chamadas encerradas/resultados, Faça
aqui o download de todos os modelos de placas disponíveis, Faça aqui o
download de todos os modelos de banners disponíveis para empresas, Faça aqui
o download de todos os modelos de banners disponíveis para ICTs"
      ["url"]=>
      string(34) "http://www.finep.gov.br/logomarcas"
      ["score_t1"]=>
      string(17) "44.03187847137451"
      ["total"]=>
      string(17) "44.03187847137451"
    }

      [...]
 }
```

Tabela 6: Exemplo de Estrutura de Retorno do Webservice

Dificuldades Encontradas

Durante o desenvolvimento da aplicação foram enfrentados alguns problemas. Alguns deles foram solucionados, outros foram desconsiderados.

1. Codificação: um dos maiores problemas foi a codificação das páginas. A maioria se encontra em *UTF-8,* mas outras são disponibilizadas em *ISO-8859* ou *latin1.* Para trabalhar os textos de forma limpa, sem erros tanto na insersão no banco de dados, quanto no retorno, algumas conversões são necessárias. Porém, nem sempre esta codificação está claramente especificada nas páginas, ou mesmo esta especificação não é facilmente identificada.
2. Formatos Proprietários: Alguns dos sites possuem páginas em linguagem *flash,* que é uma linguagem proprietária, de dificil leitura, assim como outros formatos.
3. Falta de permissões de usuário no servidor disponibilizado para concluir algumas configurações, dependendo de terceiros para efetuar as alterações.

Resultados

Para avaliar os resultados, foram realizadas algumas pesquisas e analisados seus resultados. Os testes realizados tem como objetivo principal, medir a efetividade das buscas. As buscas devem retornar as páginas contendo as palavras, em uma ordem que considere a importância de cada termo dentro do texto inteiro – uma página.

Teste 1: Para o primeiro exemplo, foi realizada a busca no endereço www.finep.gob.br por uma composição dos termos comuns “pesquisa” e “sociedade”, onde se espera encontrar o endereço “http://www.finep.gov.br/imprensa/noticia.asp?cod_noticia=2606”, de onde foram tiradas as palavras.

Resultado (3 primeiros):

url	score_t1	score_t2	total
http://www.finep.gov.br/comissaodeetica	13.82	4.28	18.1
http://www.finep.gov.br/imprensa/noticia.asp?cod_noticia=2606	4.72	11.64	16.36
http://www.finep.gov.br/imprensa/noticia.asp?cod_noticia=2532	7.9	6.24	14.15

Tabela 7: Resultado Teste 1

Analisando:

No primeiro resultado, conferindo o conteúdo *HTML* da página, a palavra “pesquisa” aparece três vezes, enquanto a palavra “sociedade” aparece apenas uma vez. Como as frases encontradas nesta página são curtas, o resultado é maior.

No segundo resultado, que é o endereço esperado, a palavra “pesquisa” aparece oito vezes, enquanto a palavra “sociedade” aparece apenas uma vez (Ilustrações 4, 5 e 6). Apesar de a contagem das palavras ser maior neste segundo resultado, como as palavras são encontradas em textos maiores, parágrafos, tornando a pontuação menor em relação a cada grupo de texto.

No terceiro resultado, cada uma das palavras ocorre duas vezes e por isso as pontuações não ficaram muito distantes. Da mesma forma como no segundo resultado, os textos são extensos o que torna as pontuações menores.

Ilustração 5: Teste 1 - Endereço testado

O objetivo da série "Debate FINEP" é criar um espaço aberto e permanente de discussão entre a Financiadora de Estudos e Projetos e a sociedade, para subsidiar a construção de ações de apoio à inovação de forma democrática, transparente e eficiente. Os eventos são abertos ao público e não é necessária inscrição prévia.

Ilustração 7: Teste 1 - Ocorrência da palavra "sociedade"

Luiz Fernando Lima Reis, diretor de Pesquisa do Hospital Sírio-Libanês. (Foto: João Luiz Ribeiro/FINEP)

Nesta terça, 12/7, na quarta edição da série Debate FINEP, o foco da discussão foram as questões da inovação na área de saúde do Brasil, um dos temas do livro "Inovações Tecnológicas no Brasil: Desempenho, Políticas e Potencial", organizado por Ricardo Sennes e Antonio Britto Filho e lançado durante o evento. Durante as apresentações e discussões, os pontos mais enfatizados foram a necessidade de maior cooperação entre academia, governo e empresas, a inovação como ferramenta de desenvolvimento e a urgência para se vencer a burocracia excessiva.

O **livro** aborda diversos aspectos da evolução das políticas públicas para ciência, tecnologia e inovação. Foram selecionados 14 artigos, incluindo um de Glauco Arbix, presidente da FINEP. Durante o debate, mediado por Rodrigo Fonseca, assessor da Presidência da FINEP, especialistas discutiram as experiências de suas instituições em relação à inovação tecnológica no Brasil. O Conselheiro da Interfarma (Associação da Indústria Farmacêutica de Pesquisa), Jorge Raimundo, explicou que a ideia da publicação, idealizada pela Associação, é "trazer à luz do debate o tema da inovação".

A importância da biodiversidade brasileira como ferramenta para a inovação foi o centro da apresentação de Antônio Paes de Carvalho, presidente da Extracta Moléculas Naturais. Segundo Antônio, a biodiversidade "é uma vantagem competitiva para a indústria nacional". Ele afirma que é urgente vencer obstáculos, como "a burocracia e a lentidão na tomada de decisões", para que pesquisas relevantes de novos fármacos, por exemplo, possam se tornar produtos no mercado. "Falta agilidade", resume. Além disso, Antônio diz que é urgente a mudança do texto da lei que rege esta área no Brasil, "que impede que moléculas retiradas da natureza sejam patenteadas, o que apenas favorece a indústria farmacêutica estrangeira", afirma.

Luiz Fernando Lima Reis, diretor de Pesquisa do Hospital Sírio-Libanês, disse que os hospitais devem investir em pesquisa em inovação tecnológica "que tenham como finalidade a melhoria da assistência, razão final de uma instituição de saúde", diz. A experiência do Sírio-Libanês, segundo ele, tem como palavra-chave a aplicabilidade. Isso significa que "os pesquisadores e médicos caminham lado a lado na busca de soluções", afirma. O hospital já tem patentes registradas decorrentes do investimento maciço em pesquisa direcionada, e é uma referência em qualidade nas Américas.

Rodrigo Fonseca falou de como a FINEP está se reorganizando para possibilitar a integração dos vários instrumentos de fomento à inovação, além da reavaliação de programas. "Estamos empenhados para um aumento significativo de demandas de projetos a serem apoiados, não só em quantidade, mas qualitativamente", disse. A FINEP está também trabalhando para conectar seus programas, de maneira mais eficiente, às políticas de desenvolvimento do País, como o PAC. "Há uma mudança de patamar no Brasil quanto à necessidade de inovação para o desenvolvimento", disse Rodrigo.

O objetivo da série "Debate FINEP" é criar um espaço aberto e permanente de discussão entre a Financiadora de Estudos e Projetos e a sociedade, para subsidiar a construção de ações de apoio à inovação de forma democrática, transparente e eficiente. Os eventos são abertos ao público e não é necessária inscrição prévia.

Ilustração 6: Teste 1 - Ocorrências da palavra "pesquisa"

Teste 2: Foi realizada a busca por SETEC (Secretaria de Educação Profissional e Tecnologia – Ministério da Educação) e edital, "setec;editar" no endereço http://www.capes.gov.br e, como esperado, por considerar a palavra "editar" uma palavra comum em muitas páginas do portal, não foi uma boa escolha. Nos primeiros trinta registros encontrados, somente a partir da décima primeira linha dos resultados a palavra "setec" obteve uma pontuação maior que a palavra "edital".

url	score_t1	score_t2	total
http://www.capes.gov.br/bolsas/bolsas-no-pais/pvns	12.15	58.81	70.96
http://www.capes.gov.br/educacao-a-distancia/pnap	12.15	33.4	45.55

http://www.capes.gov.br/bolsas/programas-especiais/toxinologia	12.15	22.34	34.49
http://www.capes.gov.br/component/content/article/48-programas-especiais/5157-programa-de-apoio-ao-ensino-e-a-pesquisa-cientifica-e-tecnologica-em-assuntos-estrategicos-de-interesse-nacional-pro-estrategia	12.15	21.79	33.94
http://www.capes.gov.br/bolsas/bolsas-no-exterior/estagio-senior	12.15	19.54	31.69
http://www.capes.gov.br/bolsas/programas-especiais/parasitologia-basica	12.15	17.8	29.95
http://www.capes.gov.br/bolsas/bolsas-no-pais/prodoutoral	12.15	15.4	27.55
http://www.capes.gov.br/bolsas/bolsas-no-pais/pvs-capes-unila	12.15	15.23	27.38
http://www.capes.gov.br/36-noticias/6292-divulgado-resultado-de-edital-para-apoio-a-compra-de-equipamentos-para-instituicoes-de-ensino-comunitarias	12.15	14.2	26.36
http://www.capes.gov.br/bolsas/programas-especiais	12.15	13.99	26.14
http://www.capes.gov.br/bolsas/programas-especiais/2332-programa-de-formacao-de-recursos-humanos-em-tv-digital-	**12.15**	**10.4**	**22.55**
http://www.capes.gov.br/bolsas/bolsas-no-pais/dinter	12.15	10.34	22.49
http://www.capes.gov.br/bolsas/programas-especiais/3148-edital-premio-systems-link-pro-multiplicar	12.15	10.17	22.33
http://www.capes.gov.br/bolsas/bolsas-no-pais	18.19	3.83	22.02
http://www.capes.gov.br/bolsas/bolsas-no-exterior/programas-estrategicos	12.15	6.59	18.74
http://www.capes.gov.br/36-noticias/6294-capes-e-fundacao-agropolis-selecionam-sete-projetos-de-pesquisa-entre-brasil-e-franca	12.15	5.29	17.44
http://www.capes.gov.br/36-noticias/6242-prorrogado-o-prazo-para-artigos-da-revista-brasileira-de-pos-graduacao-sobre-amazonia	12.15	5.23	17.38
http://www.capes.gov.br	12.15	5.17	17.32
http://www.capes.gov.br/#busca	12.15	5.17	17.32
http://www.capes.gov.br/#	12.15	5.17	17.32
http://www.capes.gov.br/#requisitos	12.15	5.17	17.32
http://www.capes.gov.br/#criterios	12.15	5.17	17.32
http://www.capes.gov.br/bolsas/programas-especiais/pbe-dpm	12.15	4.9	17.05
http://www.capes.gov.br/36-noticias/6293	12.15	4.31	16.46
http://www.capes.gov.br/concurso-publico-capes-20122013	12.15	0	12.15
http://www.capes.gov.br/cadastrodediscentes	12.15	0	12.15
http://www.capes.gov.br/cadastrodediscentes/2164	12.15	0	12.15
http://www.capes.gov.br/servicos/sala-de-imprensa	12.15	0	12.15
http://www.capes.gov.br/36-noticias/6289-capes-realiza-2o-encontro-nacional-do-parfor	12.15	0	12.15
http://www.capes.gov.br/editais/abertos	12.15	0	12.15

Tabela 8: Resultados Teste 2

Conclusão

Para este trabalho foi desenvolvido um sistema web, disponibilizado por meio de um *webservice,* de busca de informações relevantes em páginas de instituições fornecedoras de recursos para pesquisa, assim como possibilita a busca de outras informações em páginas não relacionadas ao tema.

Foi possível separar os conteúdos de cada página, assim como o conteúdo da maioria dos arquivos texto disponibilizados nestas páginas, para que fosse possível realizar as buscas.

As buscas puderam ser melhoradas com a modificação de recursos padrão do banco de dados, como a lista de palavras *stopwords*.

Limitações e Trabalhos Futuros

- Problema dos falso-positivos nos resultados das consultas full text - podem ser usadas técnicas de clusterização, além de possibilitar um maior nível de confiança no resultado obtido;
- Não foi tratada a submissão de formulários para buscas mais profundas - submissão inteligente de formulários de busca encontrados nas páginas utilizando técnicas de busca para Deep Web;
- Não foi limitado acesso ao *webservice* para mais segurança – adicionar autenticação no webservice;

Referências Bibliográficas

HEYDON NAJORK (1999) Allan Heydon, Marc Najork. Mercator: A scalable, extensive Web crawler, Compaq Systems Research Center, Palo Alto, Califórnia/EUA.

BRAGA HENRIQUE BORGES GOMES (2011) Ednei Braga, Paulo Henrique, Thiago Borges, Wallace Gomes. WebServices: Conceitos e Práticas de Desenvolvimento de Aplicações Distribuídas. Centro Universitário de Desenvolvimento do Centro-Oeste, Luziânia, Goiás, Brasil

GOOGLE (2013) Suporte Google Webmaster.
Disponível em:
<http://support.google.com/webmasters/bin/answer.py?hl=en&answer=1061943>
Acesso em: 29 jan. 2013.

PHP.net (2013) Portal de referências e documentação do PHP
Disponível em: <http://www.php.net/manual/pt_BR/intro-whatcando.php>
Acesso em: 31 jan. 2013.

IBM Developer Works (2013) Comunidade de desenvolvedores gerenciada pela IBM.
Disponível em: <http://www.ibm.com/developerworks/br/library/os-php-zend1/>
Acesso em: 31 jan. 2013.

Xpdf (2013) A PDF Viewer for X
Disponível em: <http://www.foolabs.com/xpdf/home.html>
Acesso em: 4 fev. 2013.

Mysql Full Text (2013) Documentação do Full Text – Busca Booleana
Disponível em: <http://dev.mysql.com/doc/refman/5.0/en/fulltext-boolean.html>

MySQL.com (2013) Portal do MySQL Community Edition
Disponível em: <http://www.mysql.com/products/community/>
Acesso em: 4 fev. 2013.

W3C (2013) World Wide Web Consortium
Disponível em: <http://www.w3c.br/>
Acessado em: 20 fev. 2013

OpenSearchServer (2013) API de ferramenta de busca de código aberto
Disponível em: <http://www.open-search-server.com/>
Acesso em: 25 fev. 2013.

SPENCER (2009) Henry Spencer. REGEX Linux Programmer's Manual.

VIXIE, GREENLAND, FERNANDEZ-SANGUINO, KASTNER (2010) Paul Vixie, Steve Greenland, Javier Fernandez-Sanguino, Christian Kastner. CRON Linux Programmer's Manual.