UNIVERSIDADE LUTERANA DO BRASIL
FACULDADE DE INFORMÁTICA
TECNOLOGIA EM PROCESSAMENTO DE DADOS
CAMPUS CANOAS

MECANISMO PARA CRIAÇÃO DE RESUMOS *"GOOGLE SUMMARIZER"*

Gustavo Tagliassuchi

Monografia desenvolvida durante a disciplina de Trabalho de Conclusão de Curso de Tecnologia em Processamento de Dados e apresentada à Faculdade de Informática da Universidade Luterana do Brasil, campus Canoas, como pré-requisito para a obtenção do título de Tecnólogo em Processamento de Dados. Orientador: Prof. Stanley Loh.

Canoas, novembro de 2002.

Universidade Luterana do Brasil – ULBRA
Faculdade de Informática
Curso de Tecnologia em Processamento de Dados – Campus Canoas

Reitor:
Pastor Ruben Eugen Becker

Vice-Reitor:
Eng. Leandro Eugênio Becker

Diretor da Faculdade de Informática:
Prof. Gilberto Fernandes Marchioro

Coordenador das Disciplinas de Trabalho de Conclusão de Curso (Campus Canoas):
Prof. Denise Salvadori Virti

Banca Avaliadora composta por: *Data da defesa: 02/12/2002.*
Prof. Dr. Stanley Loh *(Orientador)*
Prof. Vania Bogorny
Prof. José Luiz Andrade Duizith

Endereço:
Universidade Luterana do Brasil – Campus Canoas
Av. Miguel Tostes, 101 - Bairro São Luís
CEP 92420-280 - Canoas/RS - Brasil

”Não é necessário saber qual é a origem do universo, mas é necessário querer saber. O ser humano não depende de um determinado conhecimento, mas depende da disposição de querer alcançá-lo”.

Autor Desconhecido

Dedico este trabalho à amada Letícia que entendeu mais do que todos a importância dele para o nosso futuro.

AGRADECIMENTOS

No desenvolvimento deste trabalho, tivemos colaborações e motivações especiais, que em diversos momentos nos auxiliaram tanto tecnicamente quanto emocionalmente. A todas estas pessoas muitíssimo obrigado.

Ao professor Stanley Loh, que incansavelmente me guiou nesta jornada, delimitando os caminhos e estimulando nas dificuldades.

Aos colegas de trabalho que entenderam os motivos de tantas ausências neste período.

Ao amigo Júlio César dos Santos Vicente pela excelente colaboração e esclarecimentos técnicos em sua vasta experiência.

Aos meus pais por me conceberem e perceberem muito cedo meu interesse por esta área e me proporcionaram acesso aos computadores.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE ABREVIATURAS E SIGLAS

.NET	Plataforma de ferramentas de programação.
Altavista	Mecanismo de busca.
Apache	Servidor *web* de código aberto.
array	Tipo estruturado de dados, um vetor.
ASP	Linguagem de programação voltada à *web*.
cache	Área reservada a dados, *buffer* de armazenamento.
CSS	*Cascade style sheets*, linguagem de formatação de dados.
framework	Padrão de códigos fonte customizável.
Google	Mecanismo de busca.
Google Summarizer	Nome dado a ferramenta desenvolvida neste trabalho.
HTML	Linguagem de Formatação de Hipertexto
http	Protocolo de transmissão de dados mais utilizado na *Internet*.
Internet	Rede que interliga milhares de computadores no mundo.
Javascript	Linguagem de programação voltada à *web*.
link	Forma de relacionamento hipertextual na Internet.
Linux	Sistema operacional de código aberto.
mecanismos de busca	Portais que indexam e permitem a busca por palavras-chave.
meta search engines	Portais que oferecem resultados de vários mecanismos de busca.
MS Windows XP	Sistema operacional da *Microsoft*.
MySQL	Sistema de gerenciador de banco de dados, de código aberto.
Northern Light	Mecanismo de busca.
Palavra-chave	Seqüência de dados a ser pesquisada.
pdf	*Portable document format*, padrão de arquivo de dados *Adobe*.
PHP	Linguagem de programação voltada à *web* e utilizada no trabalho.
portal	*Web site* com diversos tipos de conteúdos e serviços agregados.
resumo	É o resultado da sumarização de textos.
search engines	Mecanismos de busca.
SOAP	Protocolo para troca de dados via chamadas remotas na Internet.
summarizer	Abreviatura de *Google Summarizer*.
template	Modelo para integração e reutilização de dados.
Teoma	Mecanismo de busca.
URL	*Universal resource locator*, ou endereço da *Internet*.
web	Abreviatura de *world wide web*.
Web API	*Application program interface* voltada à utilização na *web*.
WSDL	Funções, parâmetros e resultados dos serviços *web*.
XML	Linguagem de marcação expansível, para troca de dados.
Yahoo	Mecanismo de busca.

RESUMO

A idéia deste trabalho foi criar um mecanismo para gerar resumos de páginas *Web*, ou seja, a partir de resultados de um determinado mecanismo de busca, extrair os dados relevantes do resultado (resumo).

Para que através desses resumos gerados o usuário possa prosseguir na sua busca por informações relevantes, e não ser guiado por processos automatizados, baseados em algoritmos genéricos, que inferem muitas vezes resultados sem à devida relevância à pesquisa.

A partir dos resumos criados pelo "*Google Summarizer*", o processo seguinte é feito através da interação do usuário com o *summarizer*, ou seja, a partir dos resumos criados e apresentados, o usuário seguirá a *URL* correspondente, de acordo com o conteúdo que estava pesquisando, eliminando referências dúbias ou com menor relevância ao seu interesse.

Palavras Chave: Google, sumarizador, resumo, internet, mecanismo de busca, pesquisa na web, SOAP, XML, criação de resumos, busca de informações, pesquisa na Internet.

ABSTRACT

The idea about doing this work is generate summaries from web pages. In this case generate it from results of web search engines, and then extract the relevant data (summary).

From this summary, the user can search of relevant data and information. Because the process is different of the data driven process of the most search engines, where in some cases generic algorithm interfere on the results without giving the relevant data to the search.

From the results created by the "Google Summarizer", the next step is a user driven one, the search and research on the summaries make the user find the relevant data and follow the appropriate URL. Eliminating in this way dubious references and the non relevant data.

Search words: Google, summarizer, summary, Internet, search engines, web search, SOAP, XML, summary creation, information search.

1 INTRODUÇÃO

Com a massificação do uso da *Internet*, se torna necessário um melhor aproveitamento das informações relevantes disponíveis. Para se encontrar mais facilmente estas informações foram criados os mecanismos de busca.

É correto também afirmar que os melhores mecanismos de busca na *Internet* (*search engines*) indexam milhares de páginas e diferentes tipos de documentos, normalmente arquivos de texto, documentos, arquivos *pdf*, imagens, etc.

"Porém sempre que é feita uma pesquisa, mesmo a resposta retornando rapidamente, em questão de segundos, o que impacta realmente é a qualidade dos resultados fornecidos (Brin e Page, 1998, p 6)".

Os mecanismos de busca retornam uma quantidade sem precedentes de informações, de maneira rápida e facilmente acessáveis. Porém, o modelo de busca oferecido pela maioria deles, limita a diversidade, a competição e a funcionalidade.

Este trabalho buscou melhorar a qualidade das informações fornecidas pelos melhores mecanismos de busca na Internet, sendo que além de fornecer seus resultados ainda elaborava um resumo do *link* fornecido para avaliação do usuário.

Ainda, dentro desses resumos criados, é disponibilizada uma busca, o que facilita ainda mais a utilização dos mecanismos e a busca por informações relevantes.

Este processo culminou com um melhor aproveitamento das buscas e uma maior facilidade para se encontrar informações relevantes na *Internet*.

"E enquanto a pesquisa na Internet se mostra mais e mais importante, a necessidade de melhores mecanismos de busca cada vez aumentará mais (Lawrence, 2000, p. 5)". Sendo assim, se justificam quaisquer esforços para facilitar a vida dos usuários de *Internet* que buscam informações relevantes.

Baseado nesta quantidade caótica de informações e dados, muitas vezes disponibilizados de maneira desconexa, se faz necessário este trabalho acadêmico, e que poderá ter continuidade a posteriori na vida profissional.

2 MOTIVAÇÃO

"Atualmente com a explosão da quantidade de informações disponibilizadas na Internet, a melhor maneira de se encontrar informações do nosso interesse é se utilizando mecanismos de busca (Ex. Google, Altavista, etc), (Ashish e Knoblock, 1997, p. 1)". Porém, a maneira tradicional, o único jeito de se visualizar as informações é navegando página por página pelos resultados fornecidos pelo mecanismo de busca.

Esta dificuldade pode ser verificada numa simples busca como a exemplificada a seguir no mecanismo *Google* pela palavra-chave "**passarela**".

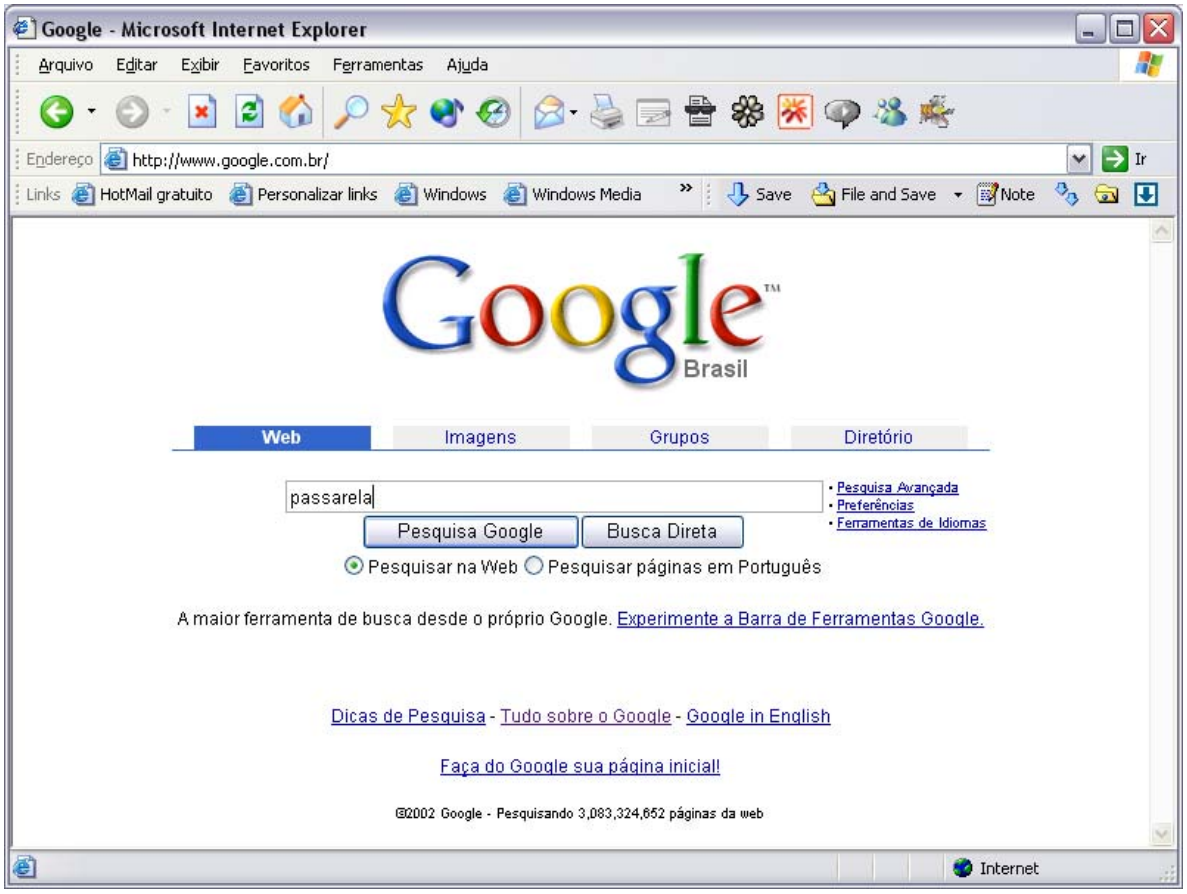

Figura 1 - Interface do mecanismo de busca Google.

Como resultado recebemos mais de 34.800 ocorrências na base de dados do *Google*. O que se percebe neste momento é que se o sentido original da busca não está claro com os resumos apresentados em cada ocorrência. Pois a palavra "**passarela**" tem diversos sentidos. Enquanto alguns textos apresentados são suficientemente explicativos, a maioria é dúbia ou sem muito nexo, logo, faz-se necessário o clique adicional, ou seja, seguir pelo *link* oferecido pelo mecanismo e visitar a *URL* indicada. Na maioria das vezes esse processo é longo e tedioso.

Um portal que possui milhares de páginas indexadas em sua base, oferece conteúdos gratuitos e pagos, necessita da máxima precisão ao oferecer resultados de pesquisa para seus clientes, pois caso o assunto desejado não seja encontrado, implica em o usuário (cliente) ir procurar em serviços concorrentes, o que impacta diretamente no seu faturamento.

O excesso de conteúdos sem valor científico ou sem embasamento espalhados pela *web* e indexados nos mecanismos faz com que uma simples tarefa de busca se torne uma tarefa demorada e muitas vezes infrutífera.

2.1 PROBLEMA

Utilizando-se o resultado do Google, após a execução de uma pesquisa, obtemos o resultado conforme a figura 2.

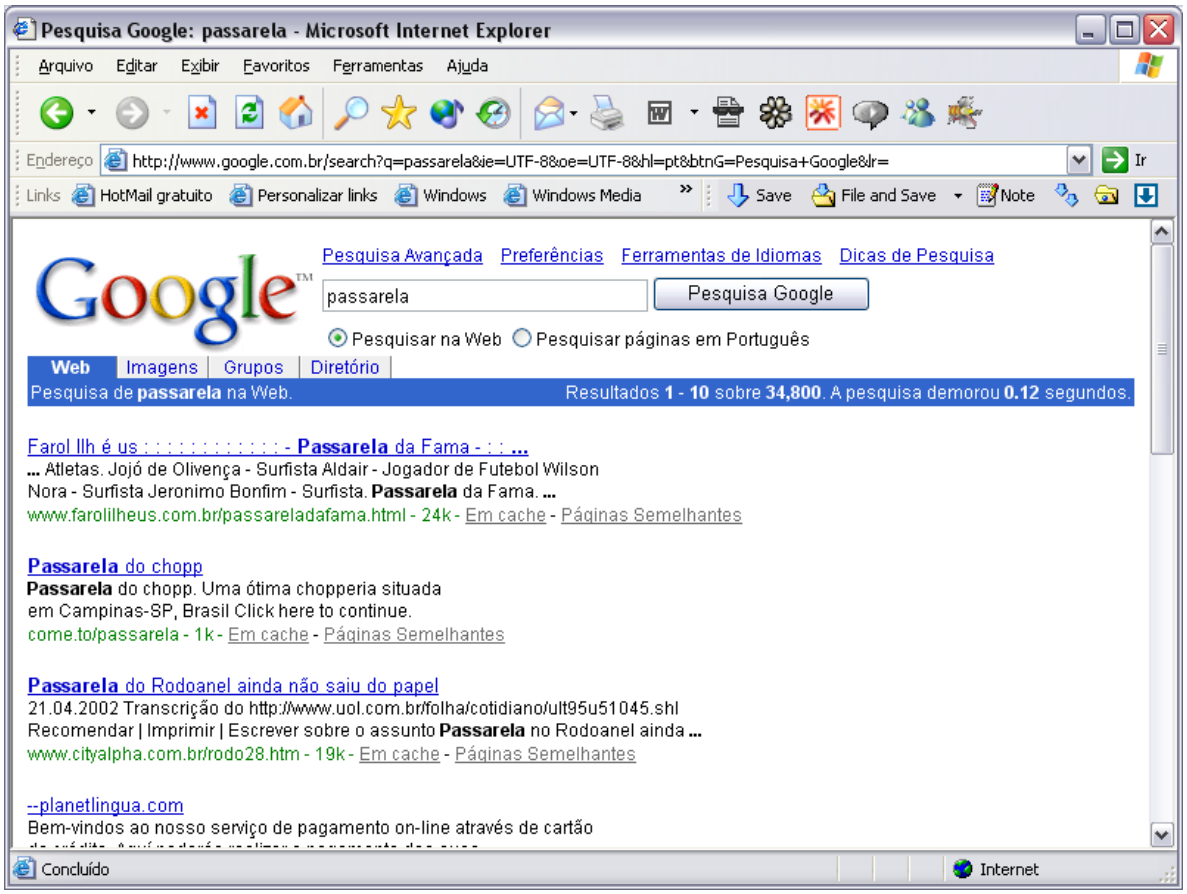

Figura 2 – Tela de retorno de pesquisa no mecanismo *Google*.

A partir daí a única maneira possível é confiar no resultado do mecanismo, e conforme o caso, clicar no *link* atrás da resposta à nossa pesquisa. O que ocorre na maior parte das vezes (dependendo do mecanismo) é que o *link* que seguimos não era o que procurávamos. Então o usuário se vê obrigado a voltar ao mecanismo atrás de outro resultado, ou de outra busca.

E esse processo é repetido inúmeras vezes, como conseqüência os usuários não encontrando o que buscam terminam normalmente suas pesquisas após a primeira página de resultados ter sido seguida. Tornando assim uma tarefa ainda mais complicada e tediosa a busca da informação.

A falta de acuidade da informação disponibilizada poderia ser facilmente contornada com a criação de um resumo, em tempo real, dos resultados disponibilizados pelos mecanismos.

3 PROPOSTA DE SOLUÇÃO

A proposta deste trabalho é criar resumos de páginas da *web* a partir de resultados de determinado mecanismo de busca (poderia ser uma fonte de dados como *Google*, *Altavista* ou até mesmo uma ferramenta de pesquisa interna de um portal como Terra ou UOL), extraindo os dados relevantes (resumo em forma de texto) do resultado, para que através desses resumos gerados, o usuário possa prosseguir na sua busca por informações relevantes.

Esses resumos serão criados em tempo real e confrontados pelo usuário, ao mesmo tempo em que os resultados do mecanismo de busca serão disponibilizados. E não o contrário, o resultado será guiado por processos automatizados, baseados em algoritmos genéricos que inferem muitas vezes resultados sem a devida relevância à pesquisa.

Para tanto, este trabalho desenvolveu um mecanismo de resumos, chamado de *Google Summarizer*. A finalidade é mostrar, em tempo real, o resultado das buscas do mecanismo Google e o resumo da página criado a partir do *link* informado, facilitando a busca por informações. Pois além de mostrar esse resumo, ainda permite ao usuário efetuar novas buscas sobre os resumos armazenados, tornando o processo guiado pelo usuário, e não guiado pelos resultados da busca de um determinado mecanismo.

A partir dos resumos criados pelo *Google Summarizer*, o processo seguinte será feito através da interação do usuário com o *summarizer*, ou seja, a partir dos resumos apresentados, o usuário seguirá a *URL* correspondente de acordo com o conteúdo a que estava

pesquisando, eliminando ainda referências dúbias ou com menor relevância ao seu interesse.

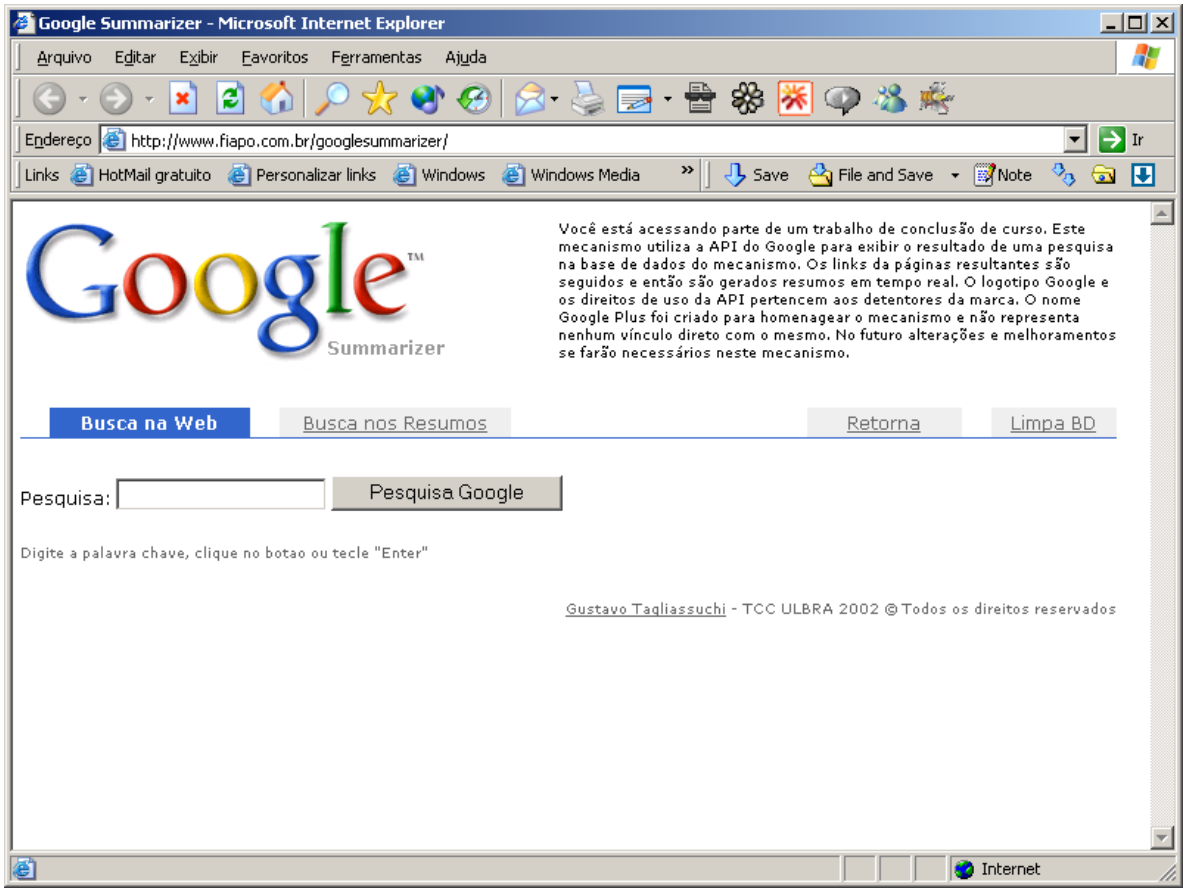

Figura 3 – Interface com o usuário do *Google Summarizer*.

3.1 IMPORTÂNCIA

Faz-se necessária à demonstração de diferenças entre o conteúdo indexado no mecanismo de busca utilizado (*Google*) e o que está realmente disponível na *URL* em tempo real.

Ainda assim é permitido ao usuário observar a relevância dos resultados e seguir a partir daí para o *link* encontrado, como faria normalmente interagindo com um mecanismo de busca, ou interagir nas informações armazenadas no banco de dados do *summarizer*, de forma a executar novas pesquisas em cima destes resumos criados e agora armazenados no seu próprio banco de dados.

3.2 TECNOLOGIA UTILIZADA

Para a criação do *Google Summarizer*, inicialmente foi utilizada uma abordagem de se criar uma aplicação em linguagem *PHP*, onde seriam importados os dados dos mecanismos selecionados (através de *templates* específicos), e posteriormente, após a criação dos resumos, seu armazenamento em um banco de dados *MySQL*.

3.3 ALTERNATIVAS ESTUDADAS

Como foi idealizado inicialmente, com a utilização de vários mecanismos de busca (*Google, Altavista, Teoma, Northern Light, Yahoo*, e outros) criou-se um obstáculo não previsto, cada *template* (entendemos *template* também como a interface entre os mecanismos e o sistema de resumos) tinha inúmeros detalhes e particularidades que dificultava o aproveitamento de códigos e necessitava da implementação de rotinas e funções específicas para cada um. Aumentando o tempo de desenvolvimento, sem se obter resultado satisfatório para a maioria deles. Tentou-se, ainda, utilizar *meta search engines*, da mesma forma insatisfatoriamente, pois os resultados variavam de acordo com a fonte pesquisada, e em alguns casos de maneira aleatória, inviabilizando novamente a criação do *template* necessário.

3.4 ALTERNATIVAS ESCOLHIDAS

Como o aspecto inicial era demonstrar o problema e sua solução, foi escolhido apenas um mecanismo (*Google*) como fonte de dados. A criação do *template* foi preterida pelo acesso direto ao banco de dados do mecanismo. Através de algumas funções disponibilizadas a desenvolvedores foi possível, mas de maneira restrita desenvolver o trabalho.

Para a utilização do mecanismo *Google* e acesso direto ao seu banco de dados (através de sua *Web API – Application Program Interface*), foi necessário ainda se incluir rotina para criação de objeto *SOAP* no aplicativo.

O *SOAP* é um paradigma cliente-servidor, construído sobre tecnologias de *Internet*, para simplificar tarefas envolvendo procedimentos e acessando objetos através de uma rede. Ele utiliza o *XML* para codificar procedimentos de solicitações (e decodificar suas respostas) num pacote ideal para transmissões através de redes, via *http*.

A *Google Web API* nada mais é do que uma licença para se utilizar a base de dados do mecanismo, desde que seja para uso pessoal, onde são disponibilizadas ferramentas e metodologias para troca de dados entre um software e a *API*.

Essa *API* foi disponibilizada pelo mecanismo *Google* como forma de manter seu desenvolvimento e comprometimento com seus usuários, e tornando sua base disponível no futuro a este tipo de utilização, massificar seu uso e se manter como referência em inovação quando se fala em busca de informações na *Internet*.

Para utilização da *Web API* do *Google* foi necessário portar a versão disponível de sua interface *SOAP (Simple Object Access Protocol)* para a linguagem *PHP*, pois o *Google* só disponibilizou versões para *Java* e para a arquitetura *.NET*. O *SOAP* nada mais é que um protocolo "leve" para troca de informações, em ambientes descentralizados e distribuídos. Ele é baseado no protocolo *XML*, que define o seu *framework*, ou a forma como as mensagens serão trocadas e suas informações transportadas na utilização do protocolo. Em síntese, o *SOAP* é um protocolo baseado em *XML* que permite aos aplicativos trocarem informações através do protocolo *http*.

O servidor *SOAP* recebe a solicitação SOAP contendo as chamadas dos procedimentos, as decodifica, executa suas funções, encapsula a resposta e envia o pacote *SOAP* de volta ao cliente que fez a requisição inicial. O cliente decodifica a resposta e utiliza o retorno da maneira mais conveniente. O processo todo é de certa forma simples, pois é totalmente baseado em padrões já existentes e solidificados, o que o torna facilmente utilizável e compreensível.

Exemplo de procedimento requisitando informações através de uma chamada via *SOAP*:

```
<?xml version="1.0" ?>
 <SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:si="http://soapinterop.org/xsd" xmlns:ns6="http://testuri.org"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <SOAP-ENV:Body>
 <ns6:getFlavourOfTheDay>
  <day xsi:type="xsd:string">monday</day>
  </ns6:getFlavourOfTheDay>
  </SOAP-ENV:Body>
 </SOAP-ENV:Envelope>
```

Figura 4 – Exemplo de requisição de dados via *SOAP*.

Exemplo de procedimento de retorno da chamada *SOAP*:

```
<?xml version="1.0" ?>
 <SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:si="http://soapinterop.org/xsd" xmlns:ns6="http://testuri.org"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
 <SOAP-ENV:Body>
 <ns6:getFlavourOfTheDayResponse>
  <flavour xsi:type="xsd:string">pineapple</flavour>
  </ns6:getFlavourOfTheDayResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Figura 5 – Exemplo de retorno de dados via *SOAP*.

Ainda por restrições quanto à criação de resumos, foi necessário criar uma metodologia diferente para representar os resultados quando é executada uma busca sobre estes

resumos armazenados, pois não foi possível identificar as frases com a devida segurança. Em muitos casos, a limpeza de código omitia a pontuação adequada, o que não permitiu utilizar este tipo de retorno ao usuário. As frases são mostradas, todas, mas com a palavra-chave pesquisada em destaque no texto. Em destaque na figura 6 o exemplo de um resultado de uma busca dentro dos resumos gerados pelo *Google Summarizer*.

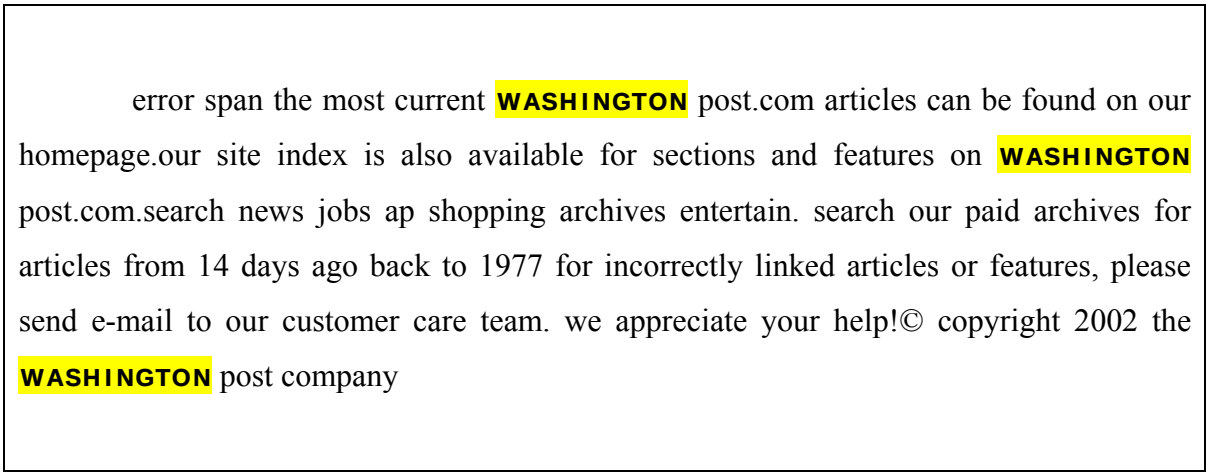
error span the most current WASHINGTON post.com articles can be found on our homepage.our site index is also available for sections and features on WASHINGTON post.com.search news jobs ap shopping archives entertain. search our paid archives for articles from 14 days ago back to 1977 for incorrectly linked articles or features, please send e-mail to our customer care team. we appreciate your help!© copyright 2002 the WASHINGTON post company

Figura 6 – Exemplo de resultado de busca nos resumos.

Pode-se observar claramente que a pontuação onde existe pode ser confundida com uma simples *URL*. Porém, fica demonstrado que o destaque dado às palavras chave encontradas nos resumos facilita a utilização do mesmo pelo usuário.

4 IMPLEMENTAÇÃO

4.1 ARQUITETURA GERAL DO *GOOGLE SUMMARIZER*

O software desenvolvido compreendeu algumas camadas: Interface em *PHP* executada em navegador padrão, funções de comunicação (*SOAP/API* do *Google*), funções de armazenamento e pesquisa em banco de dados *MySQL*, funções de limpeza de código *HTML* e outras *tags* desnecessárias e a função de sumarização.

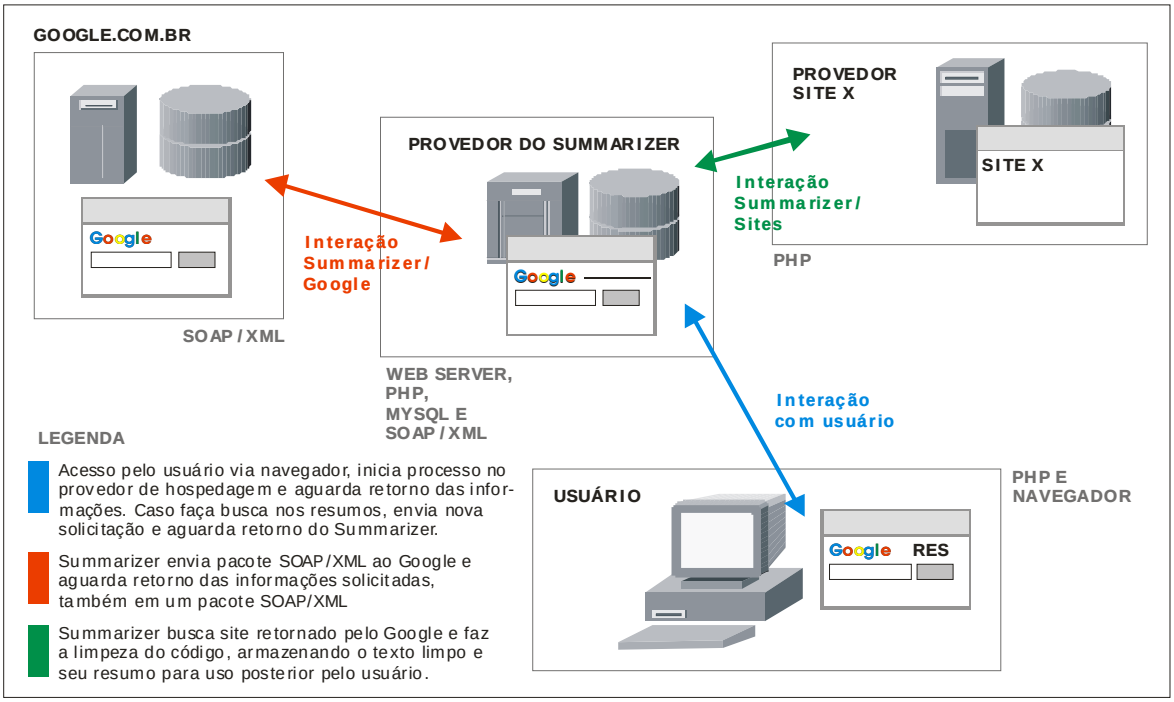

Figura 7 – Modo de funcionamento do *Google Summarizer*.

4.1.1 Interface do *Google Summarizer*

A interface do *software* foi criada com elementos gráficos semelhantes à interface do mecanismo *Google*, de maneira a facilitar a experiência do usuário, com *links* específicos para as duas funções principais, **Busca,** no *Google* e **Busca nos Resumos**. A escolha da linguagem de integração *PHP* se deveu pela familiaridade com a mesma e com a facilidade em que se encontra documentação na Internet, e ainda por ser uma linguagem aberta e utilizada por milhares de desenvolvedores mundo afora.

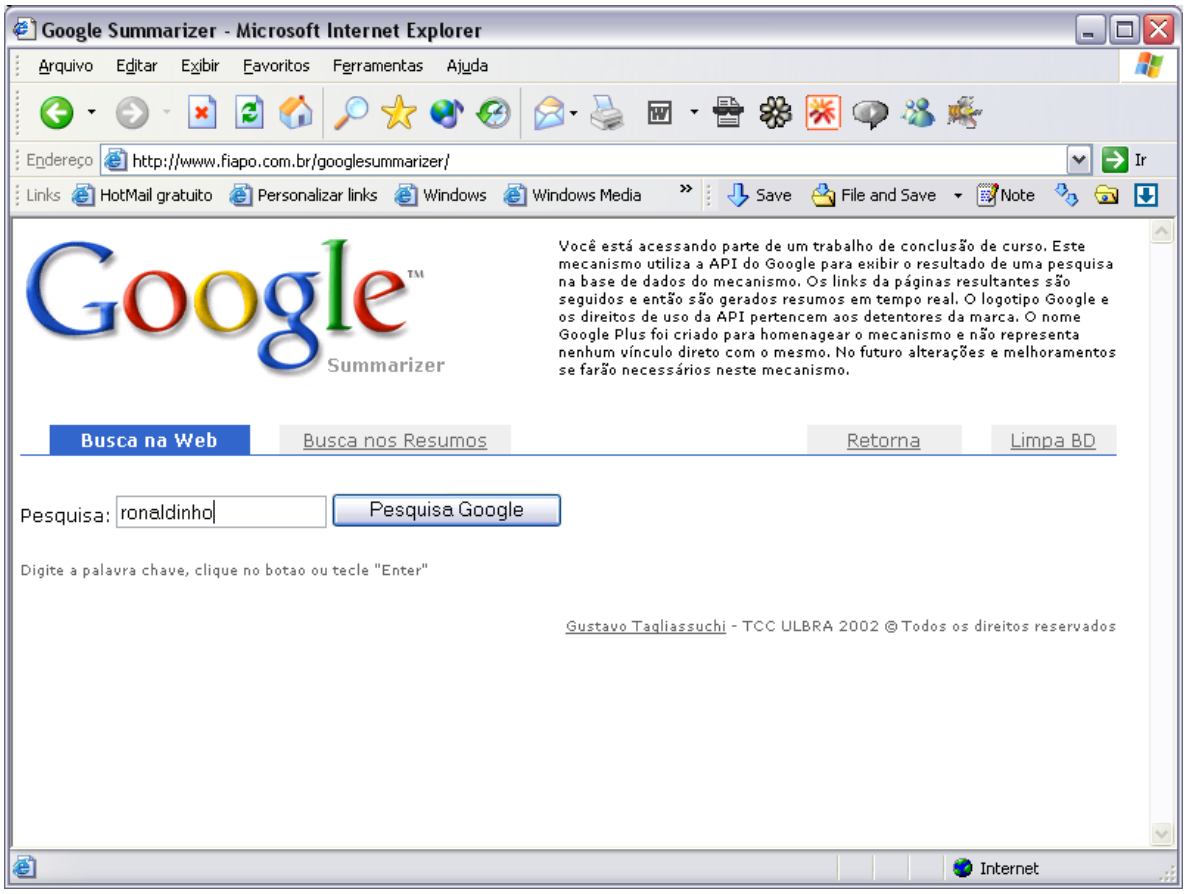

Figura 8 – *Interface* gráfica com o usuário.

4.1.2Funções de comunicação

A utilização do *SOAP* e da *API* do *Google*, permitiu a criação dos blocos *XML* de troca de informações. Porém além da criação de uma interface em *PHP* para este fim, conforme citado anteriormente, também foi necessário o cadastramento no *site* do mecanismo a fim de se obter o registro e a "*license key*". Essa chave é utilizada para validar a permissão da utilização da base de dados do mecanismo.

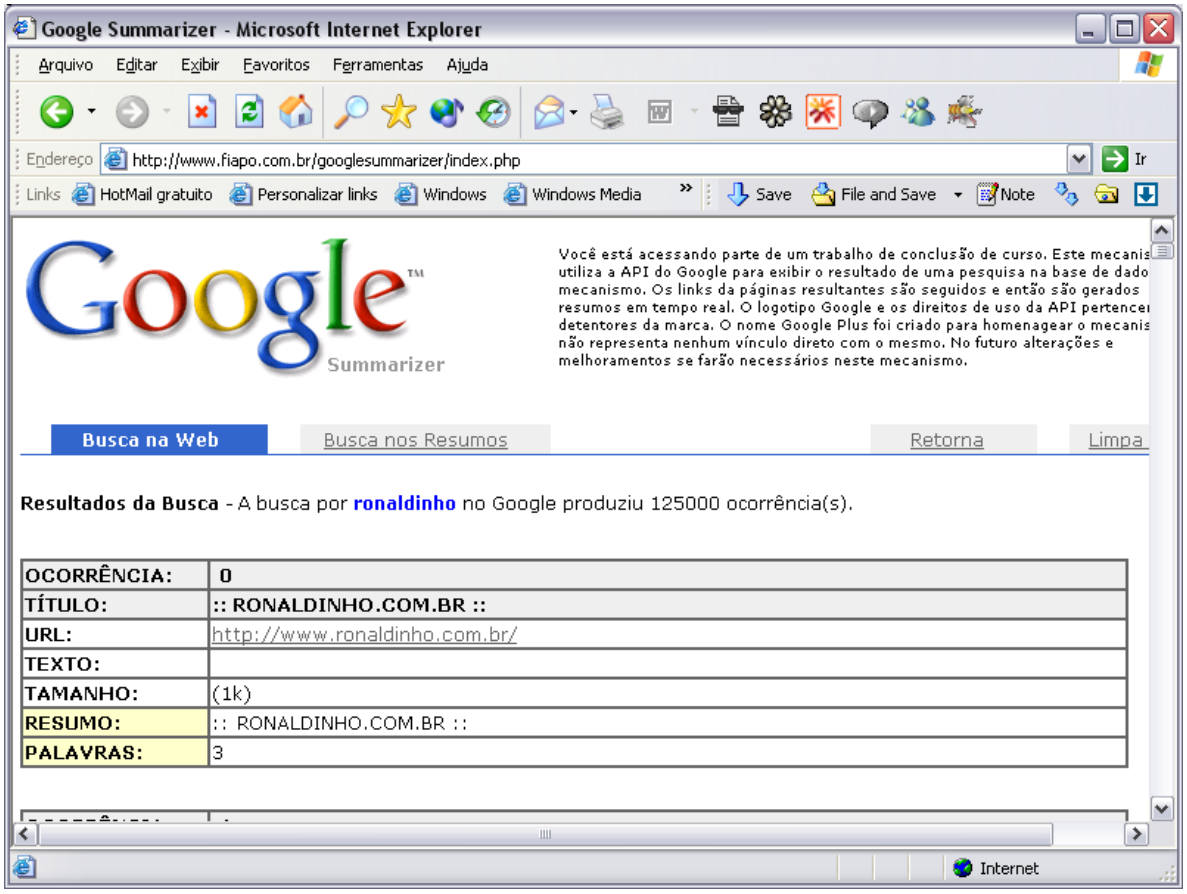

Figura 9 – Exemplo de busca efetuada.

Sua utilização ainda é restrita, permitida somente para ambientes domésticos, e tendo limitações técnicas como: possibilidade de se executar apenas 1000 buscas/dia na base do mecanismo, somente retornar 10 resultados de cada busca, não permitir codificar os textos de entrada e saída conforme o idioma utilizado, entre outras. A limitação dos 10 resultados em cada busca foi contornada.

Por outro lado é maleável, aceita as chaves de busca avançadas utilizadas no mecanismo *Google* normalmente, permite selecionar a busca por idioma, filtrar resultados para não duplicar as referências e ainda utilizar filtros para conteúdo adulto.

Os métodos disponíveis são: Busca no *Google* (utilizado), Pegue a Página do *Cache* (retorna a página armazenada no *cache* do *Google*, não utilizado) e Faça uma Sugestão Ortográfica (não utilizado).

4.1.3 Funções de armazenamento e pesquisa

Os dados retornados pelo Google, em formato *XML*, recebidos em um *array* específico, são armazenados em um banco de dados *MySQL* de tabela única, com a seguinte estrutura:

```
cod,int(11),,PRI,NULL,auto_increment
query,varchar(200),,,,
title,varchar(200),YES,,NULL,
URL,varchar(200),,MUL,,
snippet,text,YES,MUL,NULL,
cachedSize,varchar(20),,,0,
results,text,YES,,NULL,
sec,int(11),,,0,
total,int(11),,,0,
resultsfull,text,YES,,NULL,
```

Figura 10 – Tabela em banco de dados *MySQL*

A tabela armazena na ordem como são apresentados os códigos de retorno (índice), a palavra-chave em questão, o título da página armazenada no mecanismo, a *URL*, o texto disponível, o tamanho da página, dois campos de controle, o que armazena o resumo e o que armazena a quantidade de palavras encontradas e o último campo, "resultsfull" que armazena todo o conteúdo (já limpo, sem *tags*) da página pesquisada.

4.1.4 Funções de limpeza de código

Para uma melhor visualização se faz necessária à limpeza do código das páginas encontradas, removendo *tags* desnecessárias de linguagens como *HTML, PHP, ASP, Javascript, CSS* entre outras, de forma que apenas o texto resultante seja armazenado e contabilizado para a criação do resumo. Para este trabalho se utilizaram algoritmos para limpeza de códigos e algumas otimizações implementadas pelo autor, pois novas *tags* são criadas com alguma freqüência, o que implica em novas mudanças nas funções de limpeza de

código. Poderia ainda se observar à remoção de todas as *tags* e sinais, armazenando apenas texto puro, mas em alguns casos não é o ideal.

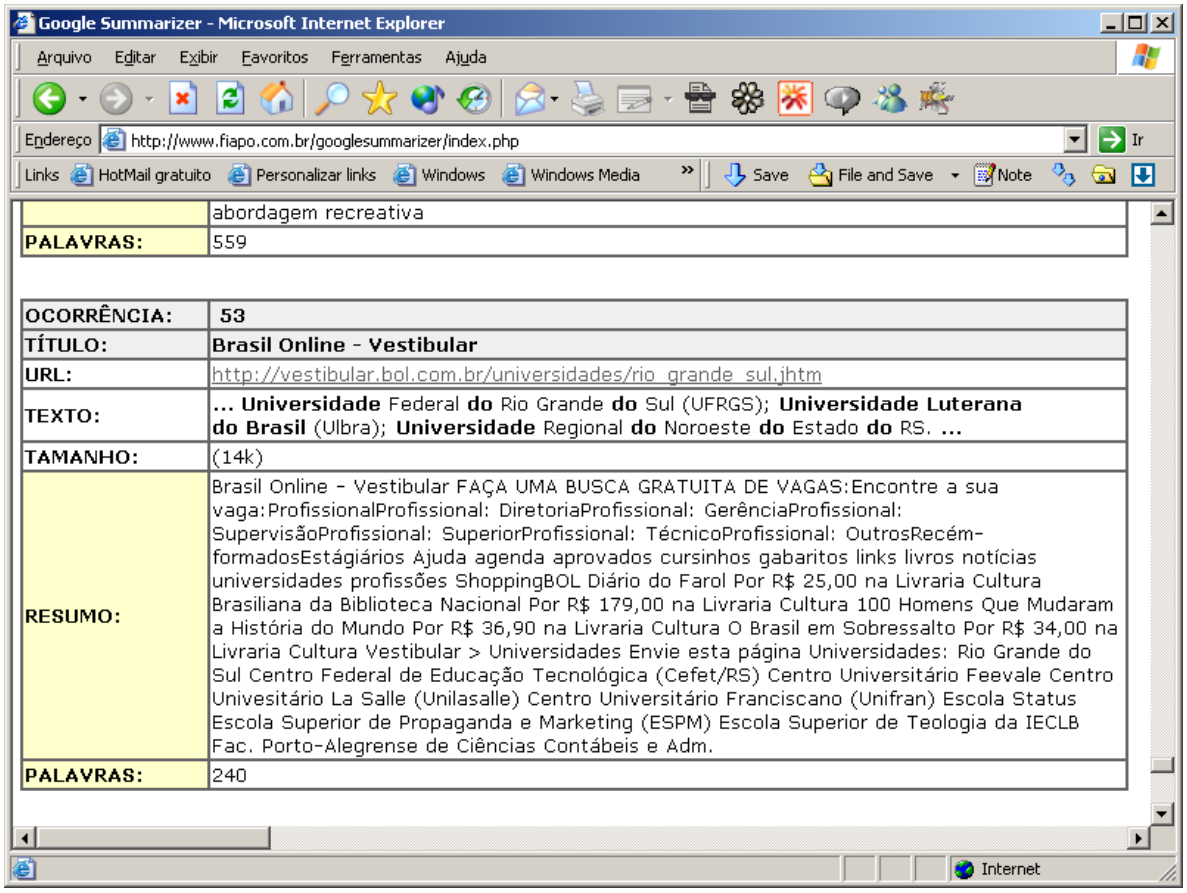

Figura 11 – Exemplo de resultado de busca

4.1.5 Função de sumarização

Não era o objetivo deste trabalho criar ou implementar técnicas avançadas e/ou automatizadas de sumarização de textos, baseada em técnicas e heurísticas existentes. Optou-se por um método simples que contempla o necessário para uma abordagem científica, a coleta das primeiras 100 (cem) palavras.

Para o aproveitamento dos resumos deveríamos ter utilizado uma abordagem simples, que selecionasse as frases onde estivesse contida a palavra-chave e as exibisse ao usuário. Porém, conforme demonstrado, não foi possível utilizar esta abordagem pela falta de indicações mínimas (pontuação ortográfica) onde se iniciaria e terminaria uma frase. Porém essa limitação foi contornada.

4.2 ARQUITETURA DO SERVIDOR DE DADOS

Como o *Google* disponibilizou apenas como fazer uso de sua *API* de acesso direto nas linguagens de programação *.NET*, e *Java*, foi necessário reescrever os módulos para sua utilização com a linguagem *PHP* e com o *SOAP*. Neste caso foi utilizado o *SOAP* como elemento de ligação entre o *XML* gerado pelo *WSDL* (*Web Service Description Language*), e que é necessário às linguagens de programação que necessitam se comunicar com a *API* do *Google*.

Esse *XML* permite descrever os serviços de rede como mensagens ou documentos, que além de se propagar, controlar seu fluxo e permitir que exista comunicação real independente das redes e protocolos utilizados, mas em padrões de comunicação da Internet, como o *SOAP e http*.

4.3 ARQUITETURA DO SERVIDOR DE HOSPEDAGEM DO *SUMMARIZER*

Para executar o trabalho se faz necessária uma infra-estrutura com as seguintes características: Servidor *web Apache*, suporte a *PHP 4*, banco de dados *MySQL*, *SOAP*, navegador e preferencialmente sistema operacional *Linux*. Para o desenvolvimento utilizaram-se as versões para *MS Windows XP* da infra-estrutura citada acima.

4.4 MODELO DE USO

A figura 12 descreve o processo de busca, retorno, coleta de textos, armazenagem de dados e possibilidade de reutilização dos mesmos para nova busca nos resumos, visando facilitar o entendimento.

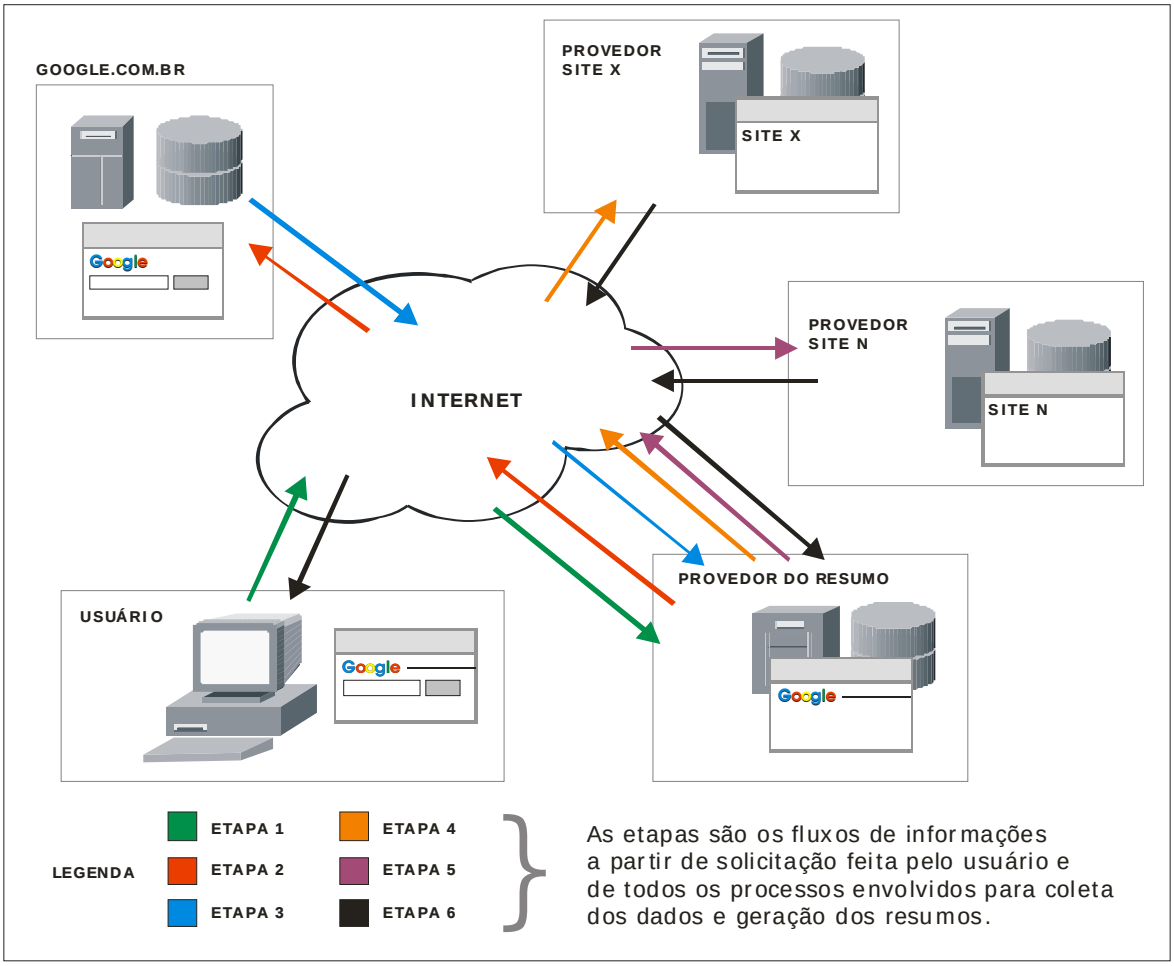

Figura 12 – Modelo de uso, com legendas.

Na figura 12 temos a implementação das etapas do sistema de resumos, pela ordem:

Etapa 1 - Legenda ●: Usuário através do seu navegador, acessa a *URL* onde está hospedado o *summarizer*, neste caso dentro de um provedor de hospedagem comum, é feita a requisição da palavra-chave;

Etapa 2 - Legenda ●: O *summarizer* armazena a palavra-chave no banco de dados, cria o objeto *SOAP*, abre uma conexão com a *API* do *Google* e envia a requisição para pesquisa.

Etapa 3 - Legenda ●: O *Google* retorna o resultado, caso exista, em grupos de 10 registros, que contém normalmente o título da página, o texto que a descreve, e a *URL*.

Antes de armazenar no banco de dados do *summarizer*, para cada um dos registros, o *summarizer* cria etapas como a 4 e 5;

Etapa 4 - Legenda ●: O sistema de resumos segue a *URL* do registro de retorno do *Google* até o *site*, captura toda a página e inicia o processo de limpeza do código, retorna via etapa 6;

Etapa 5 - Legenda ●: Outra instância idêntica da etapa 4. Sendo repetidas N vezes, tantas quantos são os registros retornados pelo *Google*;

Etapa 6 - Legenda ●: Já com o texto capturado e limpo, é feito então o resumo, a quantidade de palavras encontradas é gravada no banco, juntamente com o resumo, fechando então cada registro da tabela descrita na figura 11.

Para a pesquisa nos resumos gravados, é feita então a busca dentro do banco de dados do *Google Summarizer*, retornando então todas as ocorrências ao usuário.

4.5 ARQUITETURA DO SOFTWARE

Para o desenvolvimento do protótipo optou-se por seguir um modelo de implementação baseado em linguagens de código aberto, e com ampla documentação disponível. Ainda assim, módulos e funções foram criados e utilizados conforme as necessidades foram surgindo, de acordo com o diagrama da figura 13.

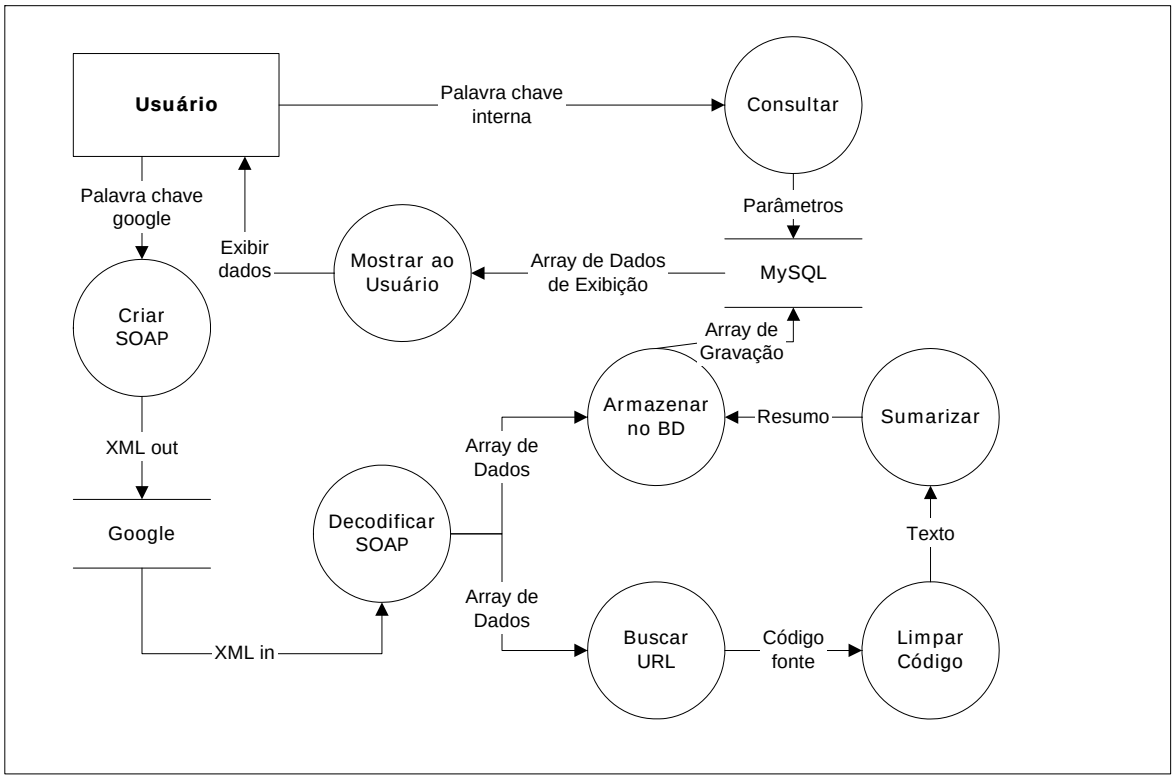

Figura 13 – Diagrama de Fluxo de Dados.

4.5.1 Módulos

Pesquisa no *Google*: A partir deste módulo são executadas todas as funções do *Google Summarizer*. Se for uma busca, inicializa o objeto *SOAP*, envia o *XML* ao *Google*, recebe o *XML* de volta, segue os dados, limpa o código, armazena no banco de dados e ainda inicializa o mesmo se for necessário, exibe os dados gerados e aguarda interação do usuário.

Pesquisa nos Resumos: A diferença com relação ao o módulo "Pesquisa" no Google é que neste caso, é utilizada a palavra-chave fornecida pelo usuário e feita uma busca diretamente na base de dados onde estão os textos completos e seus resumos. Ainda ativa uma função de exibição diferente para mostrar os resultados da busca.

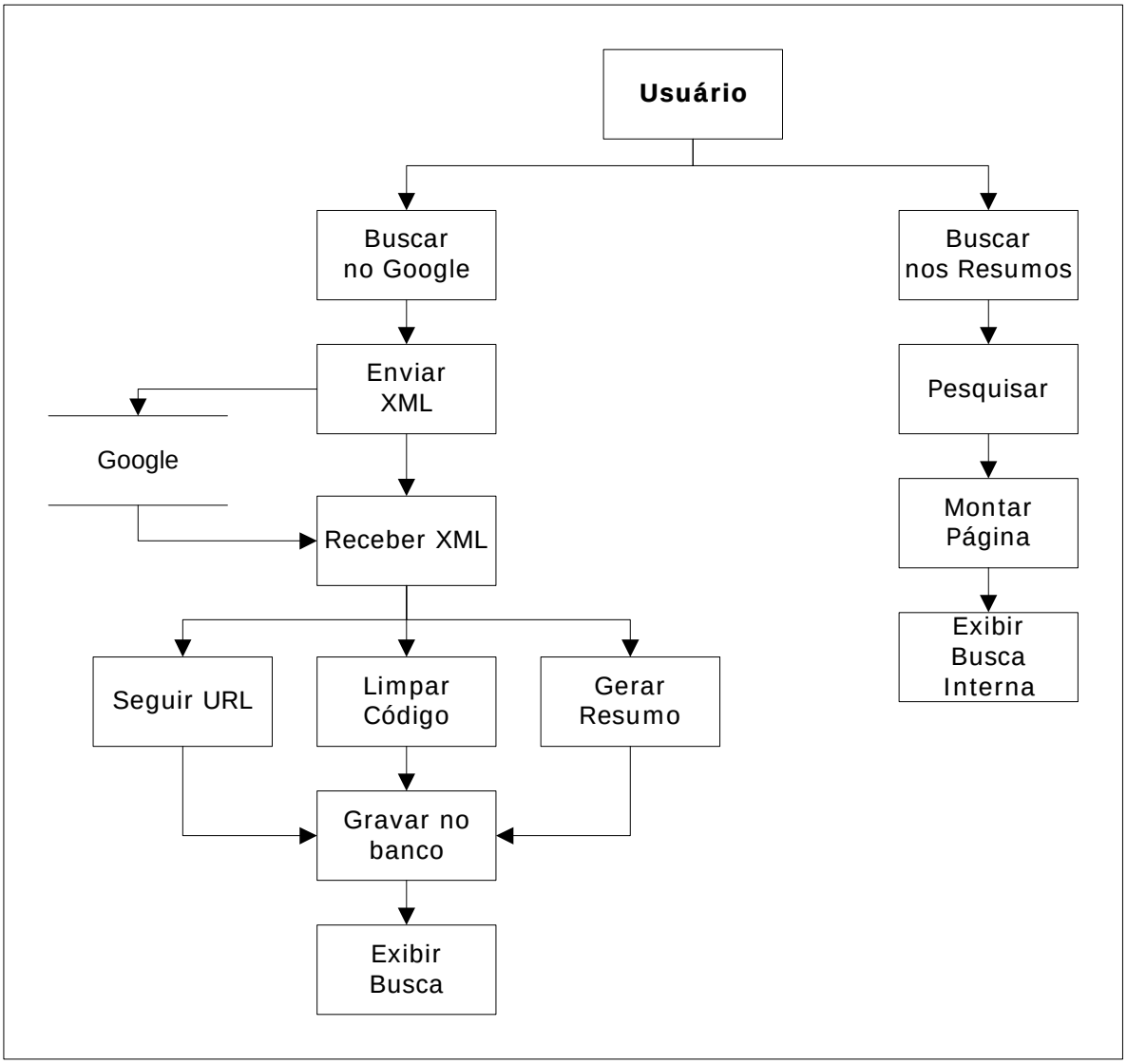

Figura 14 – Diagrama de Estrutura.

4.6 INTERFACE COM O USUÁRIO

Conforme descrito anteriormente, a *interface* com o usuário se dá através de diálogos simples, conforme a figura 8, onde quando é executada a busca no *Google*, obtivemos as respostas organizadas com a seguinte estrutura:

OCORRÊNCIA:	**3**
TÍTULO:	**Bill Clinton at ReadingChair.com**
URL:	http://www.readingchair.com/billclinton.html
TEXTO:	Click here for **Bill Clinton** at ReadingChair.com, an eclectic set of special interest pages! Main **...** Welcome to our **Bill Clinton** page. Bookmark **...**
TAMANHO:	(27k)
RESUMO:	**bill clinton** at readingchair.com main page links our online mall how to use this site shipping options customer service payment options, security, and privacy other sites by adapt, inc. amazon.com buying gifts returns policy your shopping cart at amazon.com books popular music classical music video toys electronics welcome to our **bill clinton** page.bookmark this page for future gift ideas! hot lewinsky scandal items - click any item to buy or read about it! the testimony of monica s. lewinsky, volume 1 (audiocassettes) hardcover books - click any title to buy or read about any book paperback books - click
PALAVRAS:	212

Figura 15 – Exemplo de tabela de retorno da busca.

Pode ser observada a delimitação na tabela, os campos Ocorrência (identifica o número do registro da ocorrência apresentada), Título (É o título da página, tal qual foi cadastrada pelo *Google* em sua base de dados), *URL* (É o link da página, também cadastrado pelo *Google*), Texto (É o texto armazenado no *Google* para descrever determinada página) e Tamanho (Espaço ocupado pela página no mecanismo *Google*). Os campos em amarelo, Resumo (É o resumo criado em tempo real, seguindo a *URL* e limpando o código da página) e Palavras (armazena a quantidade de palavras encontrada na página, antes de efetuar o resumo).

Percebe-se quando se realiza uma busca sobre os resultados armazenados, a necessidade de mostrar claramente ao usuário todas as ocorrências de sua palavra-chave de busca, assim como o texto completo onde podem ocorrer. É munido destas informações que o usuário vai interagir, executar novas buscas sobre o resultado, seguir o link oferecido - e agora identificado como relevante - ou ainda executando uma nova pesquisa no mecanismo *Google*.

Caso seja executada uma busca nos resumos, obtemos a seguinte estrutura:

OCORRÊNCIA:	**1**
TÍTULO:	**Bill Clinton at ReadingChair.com**
URL:	http://www.readingchair.com/billclinton.html
TEXTO:	Click here for **Bill Clinton** at ReadingChair.com, an eclectic set of special interest pages! Main **...** Welcome to our **Bill Clinton** page. Bookmark **...**
TAMANHO:	(27k)
TEXTO COMPLETO:	bill clinton at readingchair.com main page links our online mall how to use this site shipping options customer service payment options, security, and privacy other sites by adapt, inc. amazon.com buying gifts returns policy your shopping cart at amazon.com books popular music classical music video toys electronics welcome to our bill clinton page.bookmark this page for future gift ideas! hot **LEWINSKY** scandal items - click any item to buy or read about it! the testimony of monica s. **LEWINSKY**, volume 1 (audiocassettes) hardcover books - click any title to buy or read about any book paperback books - click any title to buy or read about any book videos - click any title to buy or read about any video clinton's angels audio tapes - click any title to buy or read about any cassettes bill clinton by m. banks no island of sanity, paula jones v. bill clinton : the supreme court on trial the testimony of monica s. **LEWINSKY**, volume 1 the testimony of monica s. **LEWINSKY**, volume 2 cd-rom - click any title to buy or read about any cd-rom. report of the independent counsel to the house of representatives books popular music classical music video toys electronics main page links our online mall how to use this site shipping options customer service payment options, security, and privacy other sites by adapt, inc. amazon.com buying gifts returns policy your shopping cart at amazon.com © 1999 adapt, inc. this vast right wing conspiracysite is owned by readingchair.com. want to join the vast right wing conspiracy?? [skip prev] [prev] [next] [skip next] [next 5] [random site] [list sites] [next site] [skip 1 site] [next 5 sites] [previous site] [join!]
PALAVRAS:	212

Figura 16 – Exemplo de tabela de retorno da busca nos resumos.

Um exemplo de busca foi executado para demonstrar as potencialidades do Google Summarizer. A partir de um teste prático executado pela "**Association for Computing Machinery, Melbourne 1998"**, entre diversos participantes de várias instituições acadêmicas em todo o mundo, foram fornecidas 10 perguntas para se observar o tempo de busca de suas respostas e a qualidade das mesmas. Para o teste prático foi selecionada uma pergunta. Em inglês originalmente: *I need a map showing the location of the Penfold's winery in Australia*. Traduzindo tem-se: Eu preciso um mapa exibindo a localização da vinícola *Penfold´s* na Austrália. Executou-se a busca pela palavra-chave "*Penfold's winery australia*" e recebemos o primeiro resultado.

Busca na Web | Busca nos Resumos | Retorna | Limpa BD

Resultados da Busca - A busca por **Penfold\'s winery australia** no Google produziu 79 ocorrência(s).

OCORRÊNCIA:	**1**
TÍTULO:	**Penfolds Wines**
URL:	http://www.penfolds.com/
TEXTO:	**...** Penfolds Grange, Penfolds News and Events, Winemakers and Vineyards, Cellaring and Service of Penfolds Wines, Visit Penfolds **Winery**, A History of Penfolds Wines,
TAMANHO:	(13k)
RESUMO:	penfolds wines
PALAVRAS:	18

Figura 17 – Resultado da busca em 2 minutos e 54 segundos.

somewhat packed) three-day tour of the hunter, take the cessnock exit off the freeway from sydney and drive through this country town en route to the vineyards. on the other side of town near the cessnock airport on allendale road is the vintage hunter wine and visitors centre. here you can pick up the very informative hunter valley wine country guide. it has the names, phone numbers, addresses and opening hours of all the wineries, accommodations, restaurants and other activities as well as a useful **MAP**. the hunter valley is actually divided into the lower hunter and upper hunter—most of the wine touring facilities are grouped around the pokolbin region of the lower hunter. when arranging accommodation, for top of the line go for len evans' tower lodge, a nouveau baronial folly with 12 distinctive luxurious rooms and more than a passing nod to santa fe in its stuccoed exterior. for those who would like to mix golf and a spa along with their wine tasting, stay at the cypress lakes resort. several particularly enticing bed-and-breakfast options include the carriages country house, olives country house and the woods. good mid-range hotels include peppers guest house and the kirketon park hotel. a good place to begin your exploration of the hunter

Figura 18 – Busca nos resumos retornou o termo pesquisado em 12 segundos.

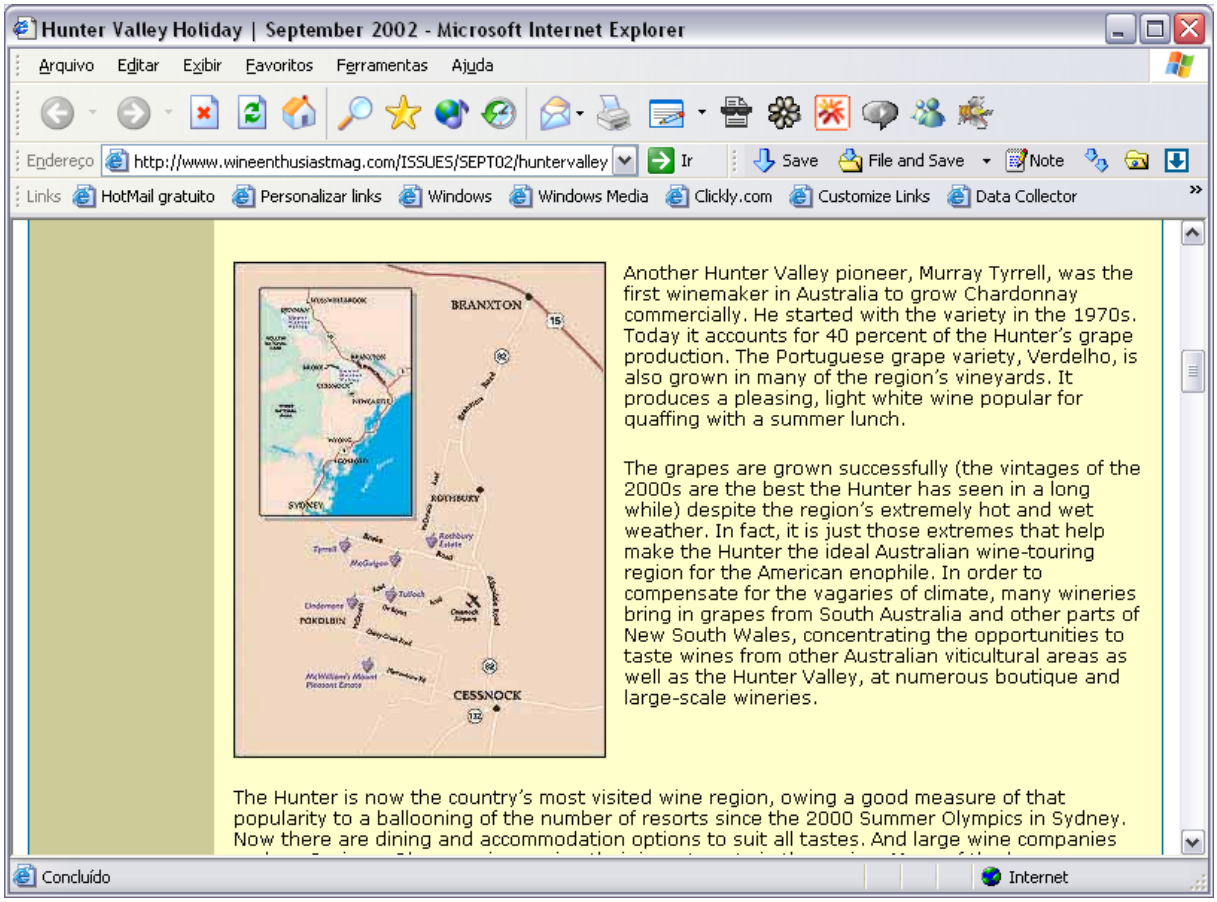

Figura 19 – Clique no *link* e leitura do texto em 20 segundos.

Logo ficou claro que o resultado foi encontrado facilmente graças à habilidade da criação de resumos e de um pouco de sorte. Por exemplo, o teste efetuado na cidade de Iowa, para buscar o mesmo resultado, levou 8 horas para ser concluído, após 10 buscas distintas e várias horas de navegação. Estes tempos distintos também se deveram a fatores como a utilização de mecanismos de busca ineficientes, falta de experiência do usuário, a

demora em se ler e interpretar todos os textos e informações recebidas dos mecanismos entre outras.

Este é um simples exemplo sobre um protótipo que não está otimizado, mas nem por este motivo deixa de ser prático e útil ao atingir seus objetivos.

5 CONCLUSÕES

A busca por informações é de extrema importância para pessoas e empresas. Pesquisar na Internet não significa encontrar exatamente aquilo o que se procura, devido à quantidade de informações e dados disponíveis.

O desafio deste trabalho foi provar que mesmo se utilizando mecanismos de busca, ainda podemos melhorar os resultados. Na primeira parte deste trabalho foi colocado o problema identificado, na segunda parte a proposta para resolvê-lo, juntamente com as tecnologias empregadas. Na terceira parte, a sua implementação e descrição de sua arquitetura e funcionamento. Ainda foram descritas as dificuldades encontradas e as soluções para contorná-las de maneira satisfatória.

Os resultados dos mecanismos de busca na Internet precisam melhorar. Fica claro com os resultados do *Google Summarizer* que as buscas ficam mais facilitadas com a utilização de técnicas simples e a criação de resumos em tempo real. E sua incorporação facilitaria a vida dos usuários da Internet.

Faz se necessário lembrar que as aplicações para este trabalho são inúmeras e não somente aquelas citadas como motivacionais. Mesmo com as limitações técnicas de velocidade e quantidade de resultados, ainda assim se percebe que diversas aplicações práticas para esta abordagem podem ser implementadas.

Como sugestões futuras, faz-se necessário ampliar a capacidade da ferramenta (novos mecanismos de busca) e também otimizar a performance do mesmo. A utilização de algoritmos mais elaborados de criação de resumos poderia ser implementada, como por exemplo: frases mais significativas, frases do assunto principal e frases de assuntos periféricos.

6 REFERÊNCIAS BIBLIOGRÁFICAS

1. ASHISH, N. e KNOBLOCK, C., **Wrapper generation for semi-structured Internet sources**, Information Sciences Institute and Department of Computer Science, University of Southern, USA, 1997.

2. BRIN, S. e PAGE, L., **The anatomy of a large-scale hypertextual Web search engine**, Computer Science Department, Stanford University, USA, 1998.

3. LAWRENCE, S., **Context in Web search**, NEC Research Institute, USA, 2000.

4. **Proceedings of the 21st International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR-98),** Melbourne, August 1998. New York: ACM Press, 1998.

5. LIMA, T., **Manual básico para elaboração de monografia**, 2ª ed. Canoas: Editora da ULBRA, 2000.

7 BIBLIOGRAFIA COMPLEMENTAR

1. Google Web API, documento online, criado e capturado em 2002. Disponível em: http://www.google.com/apis/.

2. SOAPx4 API Documentation, documento online, capturado em 2002. Disponível em: http://dietrich.ganx4.com/nusoap/index.php.

3. World Wide Web Consortium, documento online, capturado em 2002. Disponível em: http://www.w3c.org.

4. Manual do PHP, documento online, capturado em 2002. Disponível em: http://www.php.net/manual/pt_BR/

5. MySQL Reference Manual, documento online, capturado em 2002. Disponível em: http://www.mysql.com/doc/en/index.html

6. SONDERLAND, S., **Learning information extraction rules for semi-structured and free text**, Dept. Computer Science & Engineering, University of Washington, USA, 1999.

7. WELD, D. S. e DOOREMBOS, R., **Wrapper induction for information extraction**, Nicholas Kushmerick, Department of Computer Science & Engineering e NETBot Inc, USA, 1997.

8. LOH, S., **Concept-based Knowledge Discovery in Texts Extracted from the Web**, ACM SIGKDD Explorations, v.2, n.1, July 2000, pp.29-39.

9. AMITAY, E. e PARIS, C., **Automatically summarizing Web sites – Is there a way around it?** Division of Information and Communication Sciences e CSIRO Mathematical & Information Sciences, Macquarie University, Australia, 2000.

10. SANDERSON, M., **Accurate user directed summarization from existing tools**, Center of Intelligent Information Retrieval, University of Massachusetts, USA, 1998.