Ferramenta DB-LiOS para Avaliação de Reuso de *links* em *WWW*

Enzo Seraphim
seraphim@icmc.sc.usp.br

Renata Pontin de Mattos Fortes[1]
renata@cc.gatech.edu

Universidade de São Paulo
Inst. Ciências Matemáticas e de Computação
Departamento de Computação e Estatística
Cx.Postal 668 – CEP 13560-970
São Carlos - SP Brasil

[1] Suporte FAPESP – 99/09829-9

Ferramenta DB-LiOS para Avaliação de Reuso de *links* em *WWW*

Resumo

A dinâmica e a flexibilidade da autoria de hiperdocumentos na *Web*, por um lado popularizam a cada dia o uso da Internet, mas por outro, propiciam que facilmente muitas informações fiquem inconsistentes. Basta uma definição errônea de um *hiperlink*[2], para que o usuário se depare com uma inconsistência e se sinta "perdido". Um procedimento comum durante o desenvolvimento de um *site*[3] é a reutilização dos componentes de *link*, seja por haver na mesma página origem mais de um *link*, ou o mesmo rótulo de *link* em diversas páginas, ou ainda vários *links* para uma mesma página destino. Como um *site*, geralmente, contém uma grande quantidade de *links*, torna-se inviável a verificação manual da reusabilidade de seus *links*. A ferramenta DB-LiOS foi desenvolvida com o objetivo de automatizar a verificação da reusabilidade de *links* de um *site* da *Web*, através de processos de extração e classificação de *links*. Com a utilização de DB-LiOS, os autores de um *site* podem obter um auxílio efetivo para avaliação da consistência de seus *links*.

Palavras-chave: avaliação de *links* da *Web*, extração de *links*, classificação de *links*

Abstract

The dynamics and flexibility of websites authoring, on the one hand, popularize the Internet usage more and more. On the other hand, they lead easily to inconsistent information. A wrong definition of a hyperlink is enough to users come across with inconsistency and then become "lost in hyperspace". A common procedure used in site development is the reuse of link components, the same source page with one or more links, the same link label in different pages, or several links direct to the same destination page. In general, a site contains a great amount of links forbidding a manual verification of links reuse. In this paper, a tool named DB-LiOS is presented. It was developed to verify the reuse of links automatically, through links extracting and classification processes. Using DB-LiOS, website authors can get an effective aid to evaluate the consistency of links.

Keywords: evaluation of Web links, links extraction, links classification

1. INTRODUÇÃO

A flexibilidade e a facilidade de uso de hiperdocumentos na *Web* têm garantido um futuro cada vez mais promissor para a utilização de sistemas de hipertexto. Porém, quando a construção de hiperdocumentos envolve a montagem de milhares de páginas e centenas de milhares de *links*, ela se torna uma atividade que pode gerar muitas informações inconsistentes. Essa inconsistência, acrescida dos problemas inerentes à utilização de hipertextos, conhecidos por "desorientação" do usuário e "sobrecarga cognitiva" [6] e [14] durante a navegação pelo hipertexto, faz com que a manutenção seja uma atividade complexa. Além disso, com um volume enorme de informações a serem analisadas, a manutenção se torna uma atividade muitas vezes negligenciada.

De fato, hiperdocumentos melhor "escritos" e estruturados diminuiríam os problemas de desorientação, sobrecarga e inconsistência [1]. Porém, observa-se que a evolução tecnológica tem propiciado uma rápida popularização que, muitas vezes, leva a hiperdocumentos mal escritos que não podem ser completamente testados nem adequadamente mantidos [5].

---

[2] *hiperlink* - termo que representa uma ligação entre páginas de um hipertexto. Neste trabalho, adotaremos simplesmente o termo *link* com o objetivo de simplificar a leitura.

[3] *site* (ou *website*) - termo que referencia um hiperdocumento totalmente localizado em um domínio de *web*, ou seja, possui todas as suas páginas com mesmo início de URL.

O problema de hiperdocumentos “mal escritos” se agrava na *Web*, devido à liberdade de autoria oferecida pela linguagem HTML por meio de *tags* identificadores, porém, os *tags* trazem também a solução do problema. Os *tags* <a> e , que delimitam os rótulos (âncoras) dos *links*, desempenham um papel fundamental com relação à interatividade das páginas e possibilitam que critérios de consistência estrutural sejam investigados [2].

Com o objetivo de reduzir a complexidade da manutenção de hiperdocumentos da *Web* muitas pesquisas têm sido realizadas. Uma proposta para que sejam adotados os paradigmas da Engenharia de Software nesta tarefa destaca que é imprescindível a avaliação e medição de atributos de um hiperdocumento [17]. Neste contexto, uma análise estrutural com base em uma classificação dos *links* de um hipertexto, segundo a reusabilidade de seus componentes, foi definida [11]. A partir dessa classificação de *links*, métricas baseadas em reuso de *links* são formuladas [9], das quais valores podem ser coletados periodicamente de forma a auxiliar na administração do processo de manutenção de um *website*.

Este trabalho apresenta DB-LiOS (*DataBase - Link Oriented System*), uma ferramenta que foi desenvolvida com o objetivo de proporcionar uma avaliação automática da consistência estrutural de *websites* através de extração e classificação de seus *links*, segundo as métricas baseadas em reuso de *links* [9]. Além disso, os seguintes critérios foram também adotados em DB-LiOS: *consistência* (regularidade da aplicação), isto é, avaliação da forma de tratamento de elementos com conceitos similares ou distintos; e *reuso*, ou seja, a reutilização de objetos e operações em diferentes contextos e para diferentes propósitos. Esses dois critérios definidos em [13], compõem a abordagem de cinco dimensões que deve ser considerada para se avaliar um hiperdocumento: conteúdo, estrutura, apresentação, dinâmica e interação.

Este artigo está organizado da seguinte forma: a seção 2 apresenta as pesquisas relativas à melhoria de qualidade de hipertextos relatadas na literatura, como motivação de se avaliar *links* em *websites*. A seção 3 descreve os casos de reuso de *links* de um *website*, os quais fundamentam a base de *links* desenvolvida em DB-LiOS. Na seção 4 são descritas as principais características funcionais e estruturais de DB-LiOS. Na seção 5 são apresentados experimentos realizados e os resultados obtidos pela utilização de DB-LiOS. Finalmente, na seção 6 são apresentadas as próximas pesquisas, bem como as conclusões deste trabalho.

2. AVALIAÇÃO de *LINKS* em *WEBSITES*

Os problemas inerentes à utilização de hipertextos identificados como desorientação do usuário e de sobrecarga cognitiva são os principais motivadores de muitas pesquisas e os responsáveis pela evolução de sistemas hipertexto. Tais problemas interferem na qualidade da interface e da interatividade dos sistemas, na estruturação dos componentes dos hiperdocumentos e de certa forma, refletem na qualidade do hiperdocumento como um todo. Muito embora o termo qualidade seja vago e subjetivo, este trabalho adota qualidade como sinônimo de “adequação para uso” (*fitness for use*), entendendo por isso, a característica do produto que contempla as expectativas e necessidades do usuário [18].

A primeiras soluções propostas aos problemas inerentes à utilização de hipertextos envolviam investimentos em análise de textos e melhorias dos sistemas de navegação disponíveis [4]. Entre os trabalhos mais representativos na área de análise de textos encontram-se os desenvolvidos por Salton et al. [27], cujo foco principal era a estruturação e recuperação automática em grandes arquivos-textos, abrangendo: um modelo de processamento vetorial como estratégia para recuperação de informações, estratégias de análise automática de graus de similaridade de textos e geração automática de produtos de hipertexto, ou seja, de hiperdocumentos (ou parte deles).

Quando se trata de textos, o conteúdo dos nós é, na maioria das vezes, uma informação resultante de processos de escrita e formatação. Por esse motivo, os sistemas para verificação de ortografia, corretores gramaticais e sistemas de processamento de linguagem natural são exemplos de ferramentas apropriadas à avaliação automática e auxílio à autoria neste nível de granularidade "fina" de informações contidas no hipertexto. Um bom processador de textos que execute as funções de composição, formatação, verificação de estilo, busca, substituição e justificação é uma ferramenta importante para a tarefa de autoria de conteúdos dos nós.

Com relação ao auxílio e evolução das características funcionais de navegação em sistemas hipertexto, tem-se notado um aumento significativo de pesquisas e de produtos com interfaces mais elaboradas para apresentação e interação nos sistemas hipertexto. Os trabalhos que se enquadram nesse campo de ação são, na sua maioria, dedicados à melhoria da interface, fornecendo recursos de navegação gráfica aos usuários [16]. Exemplos dessa abordagem incluem: diagramas gráficos de visão-geral gerados automaticamente e diagramas gráficos de visão-geral de contextos específicos gerados manualmente [24]; visões de *web* do Intermedia (*Web Vision*) [31]; visões de olho-de-peixe (*fisheye*) [12]; agentes de conhecimento [29]. A avaliação dos sistemas hipertexto com tais recursos é muitas vezes realizada com auxílio de técnicas específicas para avaliação de usabilidade da interface do sistema hipertexto.

Um método proposto por Bernstein [3] para descobrir automaticamente os *links* a partir do conteúdo entre unidades relacionadas em um hipertexto de monografias, enfatiza que um hipertexto é inútil caso não forneça aos leitores uma coleção significativa de *links*. Os autores de hipertextos, no entanto, freqüentemente lamentam a dificuldade que experimentam em descobrir, explicar, e editar os *links* de um hiperdocumento. Halasz [14] também reforça a queixa com relação à edição dos *links*, como uma atividade maçante. Por sua vez, a atividade de manutenção se torna ainda mais crítica, com grande volume de informações a serem analisadas, e muitas vezes é negligenciada [17].

Sob uma perspectiva mais abrangente, além de um maior investimento em análise do conteúdo dos nós e aprimoramento das características funcionais de navegação disponíveis, tem havido uma maior preocupação com a modelagem do domínio de informação (o projeto) do hiperdocumento. Um modelo recente proposto por Schwabe et al. [28] consiste de sistemática de projeto baseada no modelo OOHDM (*Object-Oriented Hypermedia Design Model*). Além de subsidiar as fases de desenvolvimento de hiperdocumentos para aplicações hipermídia, nele existe um diferencial importante que é a visão orientada a objetos na fase de modelagem das informações contidas no hiperdocumento. Deve-se observar que o OOHDM já considera a implementação de hiperdocumentos em HTML, ou seja, de um servidor de WWW *(World-Wide Web).*

A *Web* juntamente com os *browsers*[4] possuem as características de sistema hipertexto, mas com particularidades. Seguindo o modelo de referência de Dexter [15] um sistema hipertexto é dividido em três camadas: camada de tempo-de-execução (*run-time*), de armazenamento (*storage*) e interna-aos-componentes (*within-component*). A camada de armazenamento é o principal foco do modelo e é onde se encontra a modelagem da estrutura básica da rede de componentes nós e *links*. Na combinação de *Web* com os *browsers* observamos que a camada de armazenamento (de estrutura) é embutida nos componentes.

Nos sistemas hipertexto em geral, a camada interna-aos-componentes, onde são alocados os conteúdos e estrutura interna dos nós, possui propostas de melhorias nos trabalhos relativos à análise de conteúdo dos documentos. Para a camada de armazenamento, onde se encontra a

[4] exemplos de *browsers* para *Web* são Netscape, Explorer

base da rede de nós e *links*, são investigadas formas para projetos de hiperdocumentos, além da modelagem do domínio de informações da aplicação. Finalmente, por meio de mecanismos mais elaborados na interface com o usuário e recursos adicionais para auxílio à navegação, a camada de tempo-de-execução, que trata da dinâmica de apresentação e interação com usuário, tem recebido atenção significativa para ganho de qualidade, principalmente com relação à usabilidade dos sistemas hipertexto.

Entretanto, por mais distintos que sejam os recursos desenvolvidos nos sistemas hipertexto e os esforços de trabalhos na melhoria de qualidade do hiperdocumento, atualmente é a *Web* que dissemina a cada dia um número maior de hiperdocumentos. Com a rápida evolução tecnológica, sua popularização, muitas vezes, leva a hiperdocumentos "mal escritos" que não podem ser completamente testados nem adequadamente mantidos [5]. Neste trabalho abordamos a avaliação de hiperdocumentos da *Web* com base na sua estruturação, pois:

- Um hiperdocumento que possua uma estruturação bem feita, pode suprir eventuais faltas de um ou outro recurso de navegação, quando transportado de um sistema hipertexto para outro [23].
- Muito embora seja possível obter facilmente os requisitos e informações sobre o processo de autoria a partir da experiência como leitor de sistemas hipertexto e de um especialista do domínio da aplicação, existe um problema maior a ser tratado nessa atividade; as pessoas não aprenderam como estruturar informações nas redes dos sistemas hipertexto da mesma forma que aprenderam e treinaram a escrever relatórios lineares através de composições escolares [25].
- A arbitrariedade na definição de *links* entre os nós permite grande flexibilidade mas, em contrapartida, tem muitas vezes, como resultado, um hiperdocumento no qual os usuários facilmente se tornam desorientados [32].

Sempre que possível, é desejável fornecer suporte à "estruturação" do hiperdocumento, para tornar mais fácil seu entendimento e organização e conseqüentemente, sua navegação. Isto pode ser feito através da identificação das estruturas fundamentais, na forma dos conjuntos específicos de dados relacionados. Com o suporte para uso de tais estruturas num sistema hipertexto emergem benefícios similares àqueles encontrados no uso de construções de programação de alto nível usadas em desenvolvimento de software.

3. CASOS de REUSO de *LINKS* em *WEBSITES*

Os *links* disponibilizados na *Web* são os elementos de interação do usuário com o sistema hipertexto (*browser*), que não são definidos na interface do sistema, mas sim no conteúdo dos nós de um hiperdocumento. Os *links*, de forma geral, definidos e embutidos nas páginas de um *website*, definem a estrutura de ligação entre as páginas por meio dos seguintes componentes: **1)** página onde se encontra o *link* ("página-origem"); **2)** rótulo que identifica a presença do *link* na página origem ("âncora"), que pode se constituir de um texto ou mesmo uma imagem; e **3)** página para onde o *link* aponta ("página-destino") que pode ser simplesmente um direcionamento para um outro ponto na própria página ou uma outra página. A partir desses três componentes de *links*, oito casos que representam as variações de reuso desses componentes foram especificados [11]:

Link 0 - *links* que contêm a página-origem diferente em relação a todas as demais páginas-origem do *site*; contêm âncora diferente em relação a todas as demais âncoras do *site*; e página-destino diferente em relação às demais páginas-destino do *site*.

Link 1 - *links* que, diante de todos os *links* que direcionam para uma mesma página-destino, contêm a página-origem diferente em relação a todas as demais páginas-origem do *site* e contêm âncora diferente em relação às demais âncoras do *site*.

Link 2 - *links* que, diante de todos os *links* que apresentam uma mesma âncora, contêm a página-origem diferente em relação a todas as demais páginas-origem do *site* e contêm página-destino diferente em relação às demais páginas-destino do *site*.

Link 3 - *links* que, diante de todos os *links* que direcionam para uma mesma página-destino com a mesma âncora, contêm a página-origem diferente em relação às demais páginas-origem do *site*.

Link 4 - *links* que, diante de todos os *links* que apresentam uma mesma página-origem, contêm âncora diferente em relação a todas as demais âncoras do *site* e contêm página-destino diferente em relação às demais páginas-destino do *site*.

Link 5 - *links* que, diante de todos os *links,* apresentam mesma página-origem com a mesma página-destino e contêm âncora diferente em relação às demais âncoras do *site*.

Link 6 - *links* que, diante de todos os *links*, apresentam mesma página-origem com mesma âncora e contêm página-destino diferente em relação às demais páginas-destino do *site*.

Link 7 - *links* que estão na mesma página-origem com a mesma âncora e direcionam para a mesma página-destino.

O Quadro 1 esquematiza, de forma resumida, esses oito casos de reuso de componentes de *links* (**Link 0** a **Link 7**) de um *website*. O quadro apresenta também, para cada um dos casos, o número de *links* possíveis de serem encontrados em cada página-origem do *site*. Dessa forma, em função da *Web* ser simplesmente um enorme repositório de páginas com os seus *links* embutidos, descritos em *tags* HTML, pode-se observar que a busca de páginas com as características de reuso de componentes de *links* requer um processamento não trivial e necessita do suporte de uma Base de Dados. Assim, a ferramenta desenvolvida neste trabalho, apresentada na próxima seção, se orientou pela criação de uma Base de Dados para armazenar e processar as informações referentes aos *links* de um *site*.

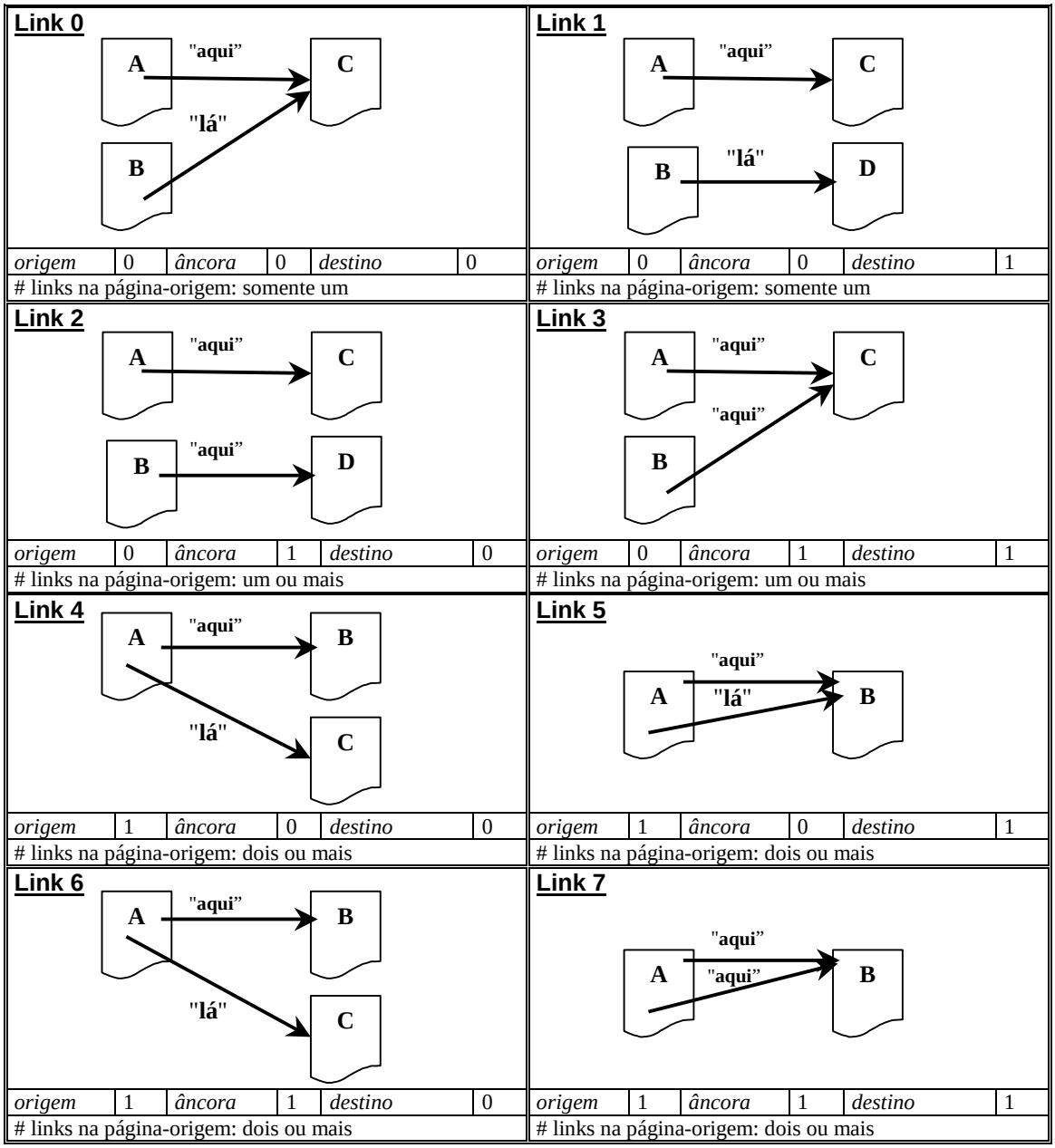
4. CARACTERÍSTICAS de DB-LiOS

Nos diversos *websites*, nota-se um crescente volume de informações disponibilizadas para navegação. Uma imensa quantidade de páginas e de *links* podem estar contidos em um hiperdocumento de um *site*, inviabilizando um controle manual da consistência de todas as informações. Para reduzir a complexidade da manutenção, um auxílio automático denominado DB-LiOS (***D**ata**B**ase - **Li**nk **O**riented **S**ystem)* foi desenvolvido, por meio de uma avaliação do reuso de *links* de um *website*, com base na classificação de *links* de Fortes [9][10][11], De fato, este auxílio visa não somente atender o enorme volume de informações a serem mantidas nos *websites*, como reduzir a complexidade da atividade de manutenção, uma vez que observa a característica de que os *links* de um *website* podem estar embutidos em qualquer local inserido nas diversas páginas do hiperdocumento.

Dois módulos funcionais de DB-LiOS realizam os principais processamentos, os quais exigem mais recursos de máquina: **a)** a extração de *links* das páginas e **b)** a classificação dos *links*. O primeiro módulo, de extração, é composto de um *crawler* [7]. O *crawler* é um programa autônomo que navega por todas páginas do *site* através de seus *links*. Os *crawlers* são utilizados para diversos fins [22], por exemplo, na coleta de palavras para formação de

catálogos de *Search Engine* [26] (como vistos em AltaVista, Yahoo, WebCrawler e Excite) e coleta de todos os documentos que formam o *site* para criar cópias espelhos do *site* [30]. Neste trabalho, a função do *crawler* em DB-LiOS é extrair as informações referentes aos componentes do *links*: página-origem, âncora e página-destino.

Quadro 1 - Casos de reuso de *links* em *website*



Obs: *Neste quadro,* 0 *significa diferente (ou seja, reuso = falso) e* 1 *significa igualdade.*

À medida que o *crawler* percorre as páginas de um *site*, são geradas as instâncias dos *links* na Base de Dados históricos, que se compõe dos dados estruturais de um *site* com seus *links* e respectivas páginas. Para o desenvolvimento do *crawler* em DB-LiOS, algumas considerações importantes foram requeridas quanto aos seguintes aspectos funcionais:

a) padronização de exclusão de URL no servidor http – observa-se um consenso entre desenvolvedores de *crawler* para que haja padronização de exclusão de URL [20]. Esta padronização estabelece que URLs que estejam no arquivo "robots.txt" localizado no

diretório raiz do *site* não podem ser requisitadas. A ferramenta DB-LiOS não considera a exclusão de URL porque a exclusão de alguma URL poderia alterar o resultado da classificação dos *links* e das páginas.

b) tempo de espera entre requisições a um servidor HTTP - recomenda-se um intervalo entre requisições para um mesmo servidor HTTP não sobrecarregar. Em [21] o intervalo sugerido é de pelo menos 1 minuto. Em DB-LiOS o usuário define qual o tempo de espera entre as requisições.

c) tempo máximo de espera para o retorno de uma requisição - em situações práticas, o servidor de HTTP pode não atender ou demorar a atender uma determinada requisição. Por isso deve-se estabelecer um tempo máximo de espera a uma requisição. Na ferramenta DB-LiOS o usuário define o tempo máximo de espera a uma requisição.

d) número máximo de requisições a uma mesma URL – esta restrição requer a definição de um número máximo de requisições feitas a uma mesma URL, para que o *crawler* não fique em *loop* eterno solicitando uma mesma URL. Na ferramenta DB-LiOS o usuário define a quantidade máxima de tentativas.

e) escalonamento das URLs - para selecionar a URL eficientemente foi implementado um escalonador baseado na política FIFO, ou seja, as primeiras URLs são as primeiras a serem requisitadas, observado o tempo de espera entre requisições (*b*).

f) tratamento do código http retornado pelo servidor - após a requisição de uma URL é retornado o código HTTP que informa o estado que se encontra a requisição. A W3C [8] estabelece uma padronização dos códigos retornados pelos servidores de HTTP. Esses códigos são agrupados em 5 grupos: Informação 1xx, Sucesso 2xx, Redirecionamento 3xx, Erro no Cliente 4xx e Erro no Servidor 5xx. Dependendo do valor do código HTTP retornado, o *crawler* de DB-LiOS realiza um tratamento.

g) definir o nome do crawler na requisição da URL - para que o servidor HTTP saiba quem é o agente responsável pela requisição da URL (*browser*, *crawler*, etc), deve-se enviar no cabeçalho da requisição, o parâmetro *UserAgent* com o nome do agente, no nosso caso DB-LiOS. Esta informação é importante para que os administradores do *site* não estranhem a grande quantidade de requisições feitas por um mesmo usuário no arquivo de log.

O *crawler* implementado na ferramenta DB-LiOS percorre somente as páginas que estão abaixo da hierarquia de um determinado diretório, pertencente ao servidor de arquivos de um *website*. Algumas vezes, observou-se que existe interesse em se avaliar os *links* para um certo conjunto de páginas restritas a um determinado subdiretório. Por exemplo, pode-se adicionar o seguinte endereço de *site* "http://www.icmc.sc.usp.br/cursos/" (observar a identificação do subdiretório /cursos/) para que DB-LiOS faça a varredura somente das páginas que estão contidas nesse subdiretório e abaixo dele.

Após extraídos os *links*, o processo de classificação dos *links* nos oito casos de reuso descritos na seção anterior pode ser iniciado. Uma vez iniciado, esse processo não deve ser cancelado pelo usuário, pois requer intensivamente os recursos de processamento e de memória. Essa exigência, embora possa ser solucionada pela utilização de máquinas mais robustas, é geralmente requerida, pois para que DB-LiOS funcione sobre qualquer SGBD (Sistema Gerenciador de Banco de Dados), o processo de classificação implementa todos os seus algoritmos na própria ferramenta, evitando assim a realização de seleções avançadas não suportadas por alguns SGBDs.

De fato, o processo de classificação, dependendo da quantidade de *links* e de páginas contidas no *site* sob avaliação, torna-se de alto custo computacional. Dessa forma, para evitar que esse processo fosse despendido para determinado *site*, a cada vez que se quisesse saber a classificação dos seus *links*, optou-se por armazenar também os resultados do processo de classificação na Base de Dados (vide Figura 1).

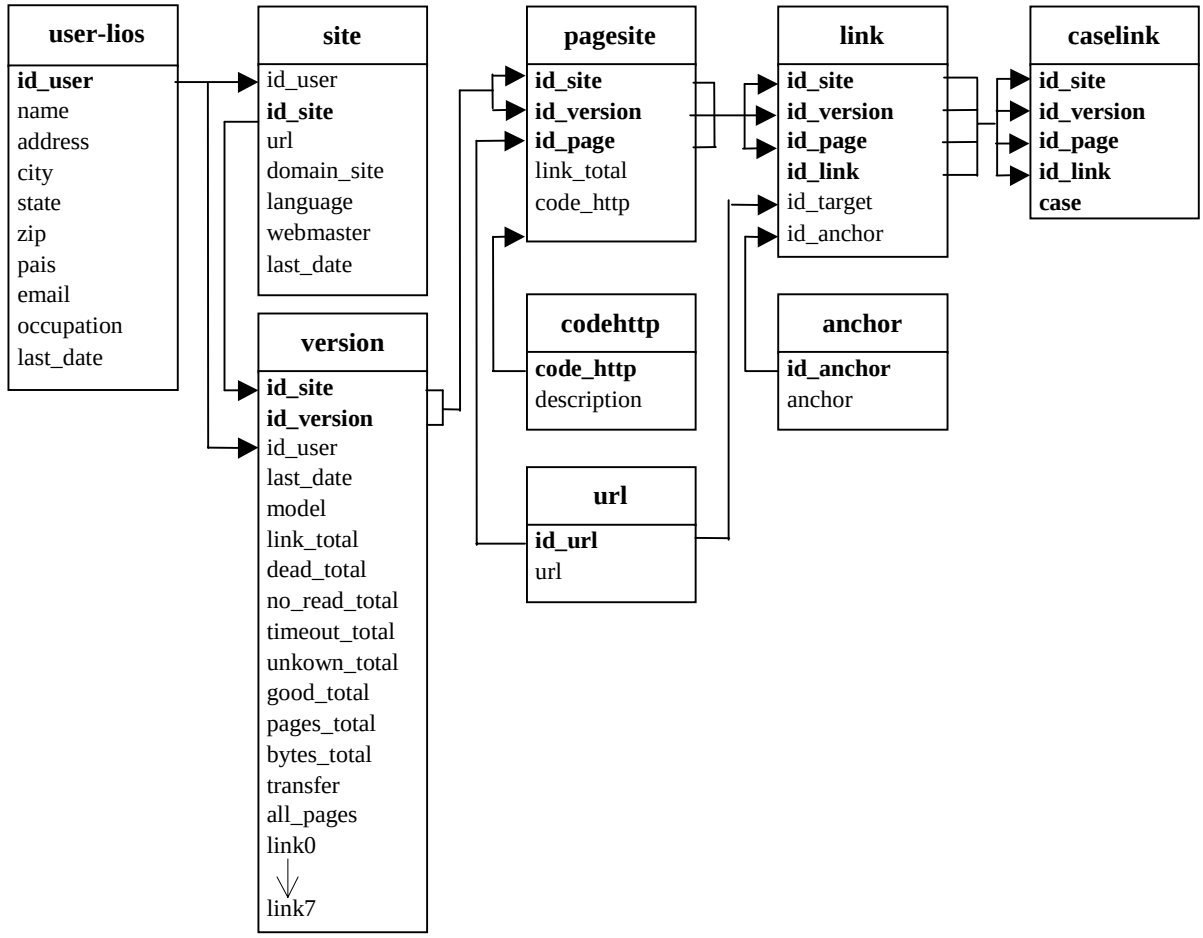


Figura 1 - Base de Dados Relacional implementada em DB-LiOS

Os algoritmos de classificação implementados verificam basicamente se página-origem, âncora e página-destino de um *link* se repetem ou não na versão do *site*. Para se verificar a repetição ou não de cada um dos três componentes de *link*, os algoritmos trabalham com estruturas de índices para cada um deles. Observa-se que a partir das estruturas de índices, a adição de um *link* em uma página ou sua remoção, a adição de uma página com *links* ou sua remoção, pode mudar a classificação de outros *links*, e reflete, portanto, a complexidade da manutenção dessas informações.

Segundo os casos de reuso de *links* propostos em [9] e [10], pôde-se também estabelecer a quantidade mínima de *links* que cada página, sob a avaliação de DB-LiOS, deve conter, de acordo com os casos de classificação que as mesmas apresentam:

- **Link 0** - a página que possui este caso apresenta somente este *link*, pois a página-origem é única diante de todo o *site*, a âncora é única diante do *site* e a página-destino é única diante de todo o *site*;
- **Link 1**, **Link 2** e **Link 3** - as páginas que possuem esses casos apresentam um ou mais *links*, pois não há reuso de página-origem.

- **Link 4**, **Link 5**, **Link 6** e **Link 7** - as páginas que possuem esses casos apresentam dois ou mais *links*, pois deve existir pelo menos 2 *links* na página-origem, já que nesses casos há reuso de página-origem.

Observa-se no Quadro 1, que para cada um dos casos, é identificado também o número de *links* possíveis de serem encontrados em cada página-origem do *site*. Essa informação foi identificada durante a elaboração dos algoritmos de classificação implementados em DB-LiOS, e enriquece a classificação previamente proposta em [9].

Conforme já mencionado, DB-LiOS também considera a freqüente evolução de um *site*, devido a mudanças nas suas páginas e seus *links*. Essas mudanças em DB-LiOS podem ser analisadas através de controle de versão do *site*. Para assegurar esses requisitos, foi elaborada a modelagem relacional da Base de Dados de DB-LiOS, e sua implementação foi desenvolvida conforme mostra a Figura 1.

Embora o SGBD que acompanha a instalação da ferramenta seja o Paradox, a Base de Dados de DB-LiOS foi definida usando tipos de dados comuns a todos os SBGDs. Portanto, nada impede a mudança da Base de Dados Local de DB-LiOS para ser executada sob qualquer outro SGBD relacional. Foram realizados testes com Microsoft Access, Interbase e PostGreSQL.

Os resultados dos diversos processamentos de classificação sobre as versões de um *site* podem ser disponibilizados posteriormente em uma Base de Dados Global, na qual todos os usuários podem compartilhar essas informações. Portanto, a ferramenta DB-LiOS implementa de fato duas visões da Base de Dados, as quais visam reduzir a complexidade da manutenção dos *websites*, que são:

Base Local → Base de Dados históricos utilizada para armazenamento de dados referentes às filtragens dos *links* das páginas dos *sites* e posterior classificação nos casos de reuso destes *links*.

Base Global → Base de Dados de Versões dos *sites* utilizada para armazenamento das informações já processadas pela Base Local. É utilizada para pesquisas dos *sites* processados pelos usuários cadastrados. A Base Global usa PostgreSQL sobre a plataforma Linux, e é compartilhada por todos os usuários da ferramenta DB-LiOS.

Para a visualização de dados contidos na Base Local (exemplificada na Figura 2), os mantenedores dos *sites* (*webmasters*) têm as seguintes interfaces implementadas em DB-LiOS:

Dados sobre *Sites* → informações gerais de identificação dos *sites*, tais como: endereço na Web (*URL*), o seu domínio de aplicação, idioma predominante na autoria das páginas, o *e-mail* do *webmaster*, a data e o nome do usuário responsável pela última atualização no *site*. Para atribuir o idioma de autoria foram utilizados como referência os encontrados na Norma ISO 639-2/1998 [19]. Para atribuição dos valores referentes ao domínio do *site* foi adotada a nomenclatura utilizada *pelo World Wide Web Corporation Consortion*, responsável pela nomenclatura do domínio ponto;

Dados sobre a(s) Versão(ões) do *Site* → informação sobre o modelo conceitual de informações utilizado para o desenvolvimento do *site*, a data e o nome do usuário responsável pela última atualização na versão sob avaliação. Apresenta os seguintes totais: nro. de *bytes* lidos, nro. de *links* bons e mortos (somente para os que apontam endereços no domínio do próprio *site*), nro. de páginas que não foram lidas, nro. de páginas que expiraram o tempo de espera do *crawler*, nro. de páginas que retornaram código de erro desconhecido, nro. de

páginas lidas, nro. de páginas mortas (páginas que não existiam no *site*), e a totalização da classificação dos *links* capturados das páginas nos 8 casos de reuso;

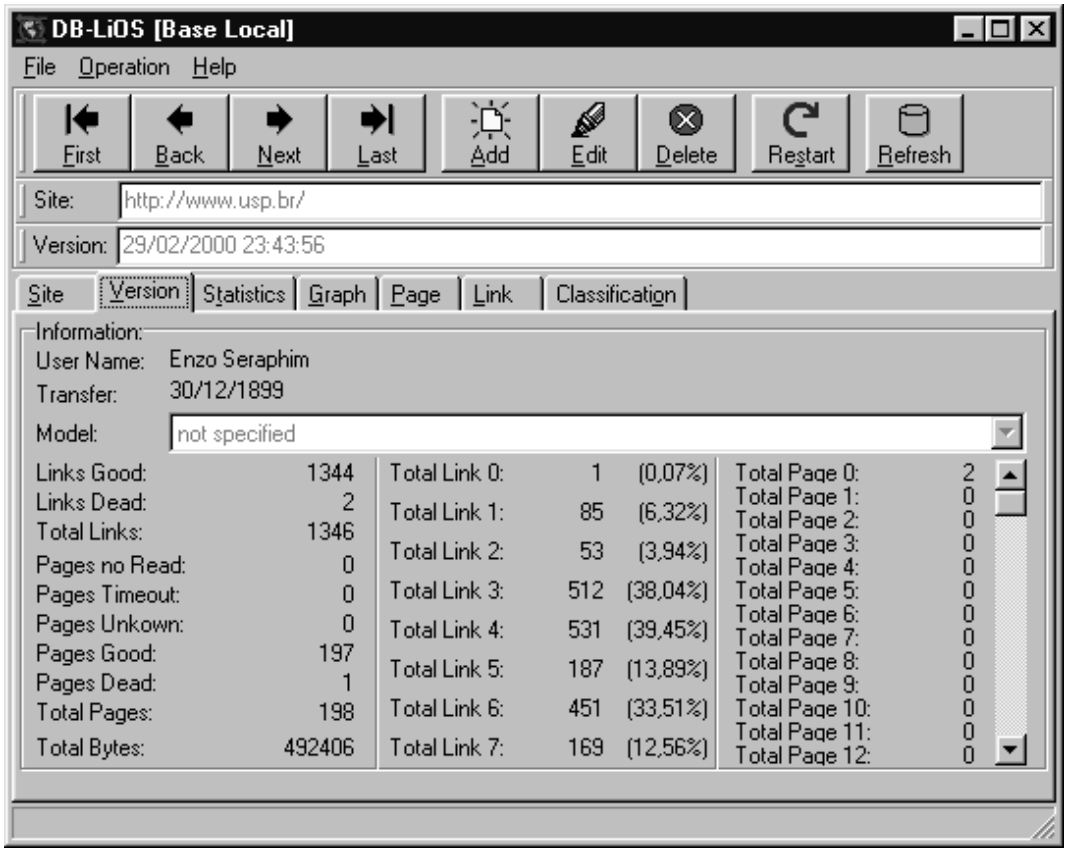


Figura 2 - Dados sobre a Versão de um *Site* em DB-LiOS.

Dados Estatísticos sobre a Versão do *Site* → informações estatísticas sobre: média de bytes por página, média de *links* por página, intervalo médio de *bytes* entre os *links*, média de páginas mortas por página, proporção de *links* mortos sobre os *links* válidos, intervalo médio de *bytes* entre *links* mortos, proporção do total de *links* sobre cada caso de reuso, média de páginas nas quais é encontrado cada caso de reuso e média de bytes percorridos entre cada caso de reuso;

Gráfico da classificação dos *links* → uma visualização gráfica da proporção de cada caso de reuso sobre o total de *links*;

Dados das Páginas → todas as páginas filtradas pelo *crawler* da versão do *site*;

Dados dos *Links* → todos os *links* filtrados das páginas da versão do *site*;

Classificação dos *Links* → todas as classes (**Link 0** a **Link 7**) de casos de reuso de *links*, avaliadas naquele *site*.

As outras funcionalidades de DB-LiOS, implementadas para suportar a manipulação da Base Local, são: **a)** incluir, alterar, excluir e selecionar um *site* ou versão de um *site;* **b)** reiniciar a filtragem das páginas: que não foram lidas, ou das que expiraram o tempo de espera do *crawler*, ou das que retornaram um código de erro desconhecido; e **c)** transferir dados dos resultados obtidos pela extração e classificação de *links* que foram disponibilizados na Base Local para a Base Global. Para que uma versão seja transferida para a Base Global, o total de páginas que não foram lidas e de páginas que expiraram o tempo de espera do *crawler* deve ser zero.

5. EXPERIMENTOS com DB-LiOS

Os experimentos iniciais realizados com DB-LiOS tiveram os resultados obtidos de 41 *sites* distintos, que são parcialmente apresentados na Tabela 1. Esses experimentos visavam validar a proposta de que a ferramenta DB-LiOS pode auxiliar a tarefa de coletar as métricas propostas em [10]. De fato, com o objetivo de proporcionar uma avaliação automática da consistência estrutural de *websites*, através de extração e classificação de seus *links*, DB-LiOS apresenta os dados coletados como valores de métricas. Além disso, esses experimentos de execução de DB-LiOS também possibilitaram uma avaliação de desempenho da ferramenta.

O requisito para a realização desse experimento foi de que os dados fossem obtidos de *websites* que possuíssem não-profissionais como seus *webmasters*, no sentido de se ter uma idéia geral de como são complexas as atividades de manutenção de consistência, principalmente quando se tem uma flexibilidade de autoria aleatória. Foram coletados e classificados os *links* de *sites* de universidades/ faculdades de ensino superior, sendo que 35 deles são instituições brasileiras e 6 são estrangeiras. Foram processadas 21.913 páginas desses *sites* no período de 26 de fevereiro a 30 de março de 2000 (vide Tabela 1).

Nº	URL	tot links	tot pags	links/pg	link0	%link0	link1	%link1	link2	%link2	link3	%link3	link4	%link4	link5	%link5	link6	%link6	link7	%link7
1	http://www.care.ac.uk/	66279	5963	14,3	35	0,0%	7459	8,7%	5810	6,6%	39934	45,7%	53497	60,7%	6958	7,0%	3035	2,4%	22727	36,7%
2	http://www.mit.edu/	53805	3683	13,6	59	0,1%	3786	7,2%	3856	6,9%	32229	60,0%	43081	80,9%	4504	8,5%	1116	2,1%	3286	6,2%
3	http://www.ime.usp.br/	9997	669	14,5	9	0,1%	726	7,6%	336	3,5%	3693	38,5%	5781	59,4%	966	10,2%	2158	22,9%	691	7,2%
4	http://www.puc-rio.br/	7231	1084	6,7	11	0,2%	1025	14,2%	469	6,5%	3352	46,4%	5385	74,6%	702	10,6%	437	6,0%	579	8,0%
5	http://www.utmem.edu/	7115	983	7,0	28	0,4%	713	10,0%	267	3,8%	3379	47,5%	6083	85,6%	808	8,6%	188	2,8%	129	1,9%
6	http://www.hokudai.ac.jp/	6822	1280	5,3	8	0,1%	1143	16,8%	655	9,6%	2722	39,9%	4929	72,2%	970	14,2%	338	4,9%	491	5,9%
7	http://www.ifo.usp.br/	6147	687	10,1	6	0,1%	451	7,3%	419	6,8%	3732	60,7%	5087	82,4%	708	11,5%	104	1,7%	265	4,3%
8	http://www.utba.br/	6127	695	8,9	7	0,1%	759	12,4%	210	3,4%	2335	38,0%	4265	77,6%	716	11,7%	205	3,3%	692	11,3%
9	http://www.fea.usp.br/	5706	553	10,3	11	0,2%	883	15,1%	106	1,9%	1821	31,4%	4721	82,7%	502	8,8%	158	2,7%	349	6,1%
10	http://www.bmc.i.cag/	5697	548	10,4	2	0,0%	282	4,9%	249	4,4%	3242	56,9%	4279	75,1%	756	13,7%	83	1,5%	536	9,4%
11	http://www.utrgs.br/	4971	748	6,7	7	0,1%	584	11,9%	209	4,2%	2090	42,0%	3755	75,5%	921	18,5%	107	2,2%	217	4,4%
12	http://www.ufes.br/	3599	321	11,2	1	0,0%	527	14,6%	324	9,0%	2295	63,8%	2885	78,6%	628	17,4%	68	1,8%	35	1,1%
13	http://www.harvard.edu/	3595	139	25,9	1	0,0%	787	22,0%	196	5,2%	1936	54,0%	2195	68,1%	322	9,0%	48	1,3%	22	0,6%
14	http://www.if.ufrj.br/	3370	710	4,3	1	0,0%	189	5,2%	862	21,6%	1280	39,1%	2681	94,2%	117	3,0%	41	1,3%	33	1,1%
15	http://www.fee.unicamp.br/	2764	364	7,6	3	0,1%	669	24,7%	88	3,2%	1480	53,7%	2410	87,5%	222	8,1%	159	5,6%	60	2,2%
16	http://www.ambiental.br/	2533	135	18,8	0	0,0%	38	1,4%	14	0,6%	2283	89,3%	2110	83,6%	386	15,2%	8	0,3%	22	0,9%
17	http://www.ic.inc.sc.usp.br/	2156	148	14,6	1	0,0%	142	6,6%	163	7,5%	551	25,5%	1833	84,9%	158	7,3%	18	0,9%	145	6,7%
18	http://www.igc.usp.br/	1892	218	8,7	11	0,6%	125	6,6%	136	7,2%	249	13,2%	1580	84,4%	55	2,9%	23	1,2%	124	6,6%
19	http://www.ffo.usp.br/	1737	318	5,6	0	0,0%	273	15,7%	267	15,5%	484	27,9%	1375	79,2%	348	20,0%	8	0,5%	8	0,5%
20	http://www.esc.sc.usp.br/	1607	278	5,8	13	0,8%	253	15,8%	36	2,2%	765	47,6%	1437	89,4%	64	3,4%	8	0,5%	89	5,5%
21	http://www.unesp.br/	1466	153	9,7	0	0,0%	85	5,7%	19	1,3%	882	59,6%	1345	90,9%	97	6,6%	8	0,5%	13	0,9%
22	http://www.unifesp.br/	1406	238	5,9	1	0,1%	168	11,2%	76	5,3%	921	65,9%	1184	84,9%	86	6,0%	22	1,6%	56	4,0%
23	http://www.usp.br/	1346	199	6,8	1	0,1%	85	6,3%	53	3,9%	512	38,0%	531	39,5%	197	12,9%	451	33,5%	169	12,6%
24	http://www.ossala.usp.br/	1150	73	15,8	0	0,0%	81	7,0%	46	4,0%	387	33,7%	885	97,9%	123	10,7%	188	16,1%	176	15,3%
25	http://www.ufrj.br/	1094	167	10,1	0	0,0%	84	8,7%	7	0,6%	810	74,7%	928	76,4%	220	20,3%	28	2,0%	9	0,8%
26	http://www.iqsc.sc.usp.br/	835	239	4,1	5	0,5%	137	14,7%	35	3,7%	319	34,1%	784	83,9%	64	6,6%	42	4,5%	16	1,9%
27	http://www.ufrs.br/	856	91	9,4	0	0,0%	30	3,5%	56	6,8%	614	71,5%	731	85,1%	55	6,4%	48	4,7%	29	3,3%
28	http://www.bioestatistica.br/	897	182	3,0	4	0,6%	53	7,0%	30	4,3%	350	50,2%	511	73,3%	102	14,6%	4	0,6%	6	0,9%
29	http://www.ufpr.br/	516	73	7,1	0	0,0%	92	17,9%	68	13,2%	70	13,6%	427	82,9%	57	11,1%	8	0,0%	27	5,2%
30	http://www.unimep.br/	501	35	14,3	2	0,4%	33	4,0%	4	0,6%	46	9,6%	182	36,3%	50	10,0%	252	50,3%	20	4,0%
31	http://www.if.sc.usp.br/	422	289	1,5	0	0,0%	89	23,5%	0	0,0%	72	17,1%	385	91,9%	32	7,6%	8	0,0%	0	0,0%
32	http://www.ft1.usp.br/	419	86	4,9	2	0,5%	65	15,6%	13	3,1%	59	14,1%	269	69,7%	93	22,2%	28	6,7%	57	13,6%
33	http://www.ft2p.usp.br/	403	52	7,8	0	0,0%	189	47,1%	22	5,5%	11	2,7%	235	58,3%	8	2,0%	154	38,2%	2	0,5%
34	http://www.pusp.br/	392	113	3,4	7	1,8%	17	4,5%	5	1,3%	20	5,2%	283	79,3%	50	13,1%	12	3,1%	14	3,7%
35	http://www.uerj.br/	393	31	9,5	0	0,0%	59	16,8%	3	1,0%	77	26,3%	184	66,2%	29	9,9%	8	2,0%	62	21,2%
36	http://www.ip.usp.br/	295	81	3,1	0	0,0%	59	23,1%	24	9,4%	59	23,1%	193	86,9%	45	17,6%	31	12,2%	16	7,1%
37	http://www.icb.usp.br/	157	37	4,2	2	1,3%	0	0,0%	1	0,6%	102	65,0%	81	58,3%	64	40,8%	8	0,0%	0	0,0%
38	http://www.eca.usp.br/	156	28	5,3	0	0,0%	27	17,4%	0	0,0%	47	30,3%	182	65,8%	48	31,0%	2	1,3%	2	1,3%
39	http://www.poli.usp.br/	109	41	2,6	1	0,9%	21	19,4%	3	2,8%	12	11,1%	87	80,6%	11	10,2%	8	5,6%	0	0,0%
40	http://www.unicamp.br/	62	10	6,2	0	0,0%	8	12,9%	0	0,0%	7	11,3%	56	90,3%	6	9,7%	8	0,0%	0	0,0%
41	http://www.fcfrp.usp.br/	47	24	2,0	7	14,9%	7	14,9%	0	0,0%	2	4,3%	35	74,5%	2	4,3%	8	0,0%	2	4,3%
	totais =	233.319	21.913	10,6	244		22.303		14.580		113.938		178.602		21.747		8.801		31.080	
	médias =			8,5		0,6%		12,6%		4,8%		38,5%		75,4%		11,8%		6,1%		5,3%
	total de links processados =	382.970				0,1%		5,8%		3,8%		29,7%		44,5%		5,7%		2,2%		8,1%

Tabela 1 - Resultados obtidos pela ferramenta DB-LiOS

Os *links* processados resultaram 382.970 classificações, ou seja, processamentos do algoritmo de enquadramento nos oito casos de reuso de *links* (basta somar os totais dos casos de **link 0** a **link 7** da Tabela 1). A diferença entre o total de *links* (233.319) dos *sites* e o total de *links* processados na classificação se deve pelo fato de que um *link* pode ser classificado em mais de um caso de reuso. Esta observação também pode ser notada nas distribuições de porcentagens de casos de *links* que não totalizam 100%, apresentadas na Figura 3.
Observou-se também que a média de *links* por páginas na soma geral foi de 10,6 e na média foi de 8,5. Esta informação representa um enorme potencial de crescimento de *links* gerados na autoria e manutenção de *websites*.

A Figura 3 resume os percentuais de cada total dos casos de reuso resultante (**link0** a **link7**) do total de *links* (233.319). Pode-se observar que o **link0** é o caso mais raro nos *sites* (0,6%) comprovando que a reusabilidade dos componentes de *link* é um fator importante durante autoria e manutenção de páginas de um *site*. Ou seja, para que um *link* seja classificado como **link0**, não pode haver reuso de seus componentes: a página-origem onde está o *link* não pode conter nenhum outro *link*, pois haveria reutilização de página-origem; a âncora associada ao *link* não pode ser reutilizada em nenhum outro *link* do *site*; e a página-destino que o *link* aponta, só deve ser apontada por ele.

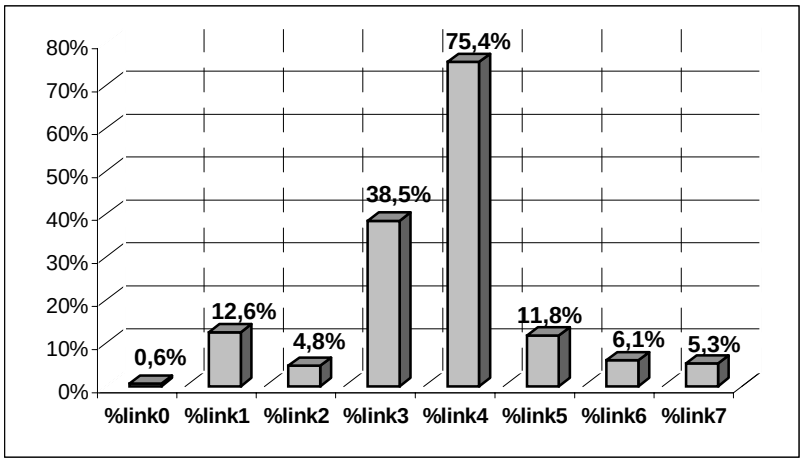


Figura 3 - Percentuais médios das classificações de casos de reuso de *links* obtidos de 41 *websites* pela ferramenta DB-LiOS

Observou-se também uma maior ocorrência na classificação dos *links* nos casos **link3** e **link4.** Os **links 3** são os que possuem mesma âncora em páginas-origem diferentes e apontam para uma mesma página-destino. Os **links 4** são os que possuem âncoras diferentes na mesma página-origem que apontam para páginas-destino diferentes.

A grande ocorrência de *links* classificados no caso do **link 4** (75,4%) indica ao usuário que âncoras diferentes em uma mesma página apontam para páginas-destino diferentes. Ou seja, geralmente usa-se âncoras diferentes para apontar conteúdos distintos. Em analogia à escrita impressa, esta estrutura equivale à de índices (conhecida também como sumário).

A segunda maior ocorrência é de *links* classificados no caso do **link 3** (38,5%), e indica que âncoras que possuem o mesmo conteúdo (ou seja, são apresentadas da mesma forma para o usuário durante a navegação, seja imagem ou textos) levam para mesma página-destino. Ou seja, quando uma página-destino tem que ser apontada novamente, geralmente usa-se a mesma âncora o que mostra consistência. Esta estruturação é análoga à de índice remissivo utilizada em redação impressa.

Vale ressaltar ainda que, o nosso objetivo com a DB-LiOS é uma avaliação de cada *site*. Existem *sites* que se utilizam de outros casos de reuso com muita propriedade e valorizam os *links* contextuais (**link 1**, **link 2**, **link 5** e **link 6**) como pode ser visto na Tabela 1. Embora esse experimento seja de um número pequeno diante da imensa quantidade de *websites* existentes atualmente, os resultados coletados mostraram que novas possibilidades de organização das informações têm sido utilizadas nas estruturas dos hiperdocumentos .

6. CONCLUSÕES

A dinâmica e a flexibilidade de autoria de *websites* propiciam facilmente muitas informações inconsistentes. Como um *site*, geralmente, contém uma grande quantidade de *links*, torna-se inviável a verificação do reuso de seus *links* manualmente. A ferramenta DB-LiOS foi desenvolvida com o objetivo de automatizar a verificação dos casos de reuso de *links* de um *website*, através de processos de extração e classificação de *links*.

A ferramenta DB-LiOS viabiliza a classificação de reuso dos elementos que compõem os *links* de *websites*. Além disso, os resultados obtidos com essa classificação mostraram que podem auxiliar os *webmasters* a verificar se os *links* de seu *site* ajudam ao usuário na navegação [10]. A consideração básica foi de que os componentes de um *link* entre páginas de um *site* são basicamente: a página-origem onde se encontra este *link*; a âncora (textos ou imagem) que identifica a presença do *link* e a página-destino do *link* que direciona para uma outra página ou um outro ponto na própria página.

A Base de Dados definida em DB-LiOS, principalmente com a característica de controle de versão, permite que sejam realizados estudos sobre a evolução estrutural do *site*. Além disso, DB-LiOS viabiliza que uma Base de Dados Global seja utilizada em trabalhos futuros no sentido de se buscar alguma uniformidade na classificação de *links* entre *sites* com o mesmo domínio.

Pode-se concluir ainda, a partir dos experimentos iniciais realizados que, os *links* classificados tanto como **link 3** ou como **link 4** são os que mais ajudam o usuário a se orientar na navegação pelas informações (páginas e *links*) do *site*, por aplicarem reuso de forma consistente. Entretanto, como são as estruturas mais semelhantes às impressas, a evolução de novas formas de organização está começando a surgir e estas formas estão sendo utilizadas nos *websites*, haja vista as ocorrências dos outros casos de *links*.

Referências Bibliográficas

[1] Begoray, J.A., "An Introduction to hypermedia issues, systems and application areas" *Intern. Journal Man-Machine Studies*, v.33, p.121-47, 1990.

[2] Berners-Lee, T. et al., "The World-Wide Web" *Comm. of the ACM*, v.37, n.8, p.76-82, 1994.

[3] Bernstein, M. "An Apprentice That Discovers Hypertext Links" In:Rizs,A.;Stritz,N.&André,J., eds. *Hypertext: Concepts, Systems and Applications* Cambridge, Great Britain, Cambridge University Press, 1990. p.212-23.

[4] Botafogo, R.A.; Shneiderman, B. "Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics" *ACM Trans. on Information Systems*, v.10, n.2, p.142-80, April 1992.

[5] Brown, P. J. Assessing the Quality of Hypertext Documents. In: European Conference on Hypertext, Versailes, França, novembro 1990. *Proceedings*. p.1-12.

[6] Conklin, J., "Hypertext: An Introduction and Survey" *Computer*, v.20, n.9, p.17-41, 1987.

[7] Cho, J.; Garcia-Molina, H.; Page L.; "Efficient crawling through URL ordering" *Proc. 7th Intern. WWW Conference*, Brisbane, Australia. April, 1998.

[8] Fielding, R. et al, "Hypertext Transfer Protocol - HTTP/1.1", *Internet Official Protocol Standards RFC-2616*, junho de 1999, http://www.rfc-editor.org/rfc/rfc2616.txt.

[9] Fortes, R.P.M.; Nicoletti, M.C., "A Family of Link Based Metrics for the Evaluation of Web Documents" *SIGLINK Newsletter*, Vol. 6, No. 3, p.21-23, 1997.

[10] Fortes, R. P. M.; Nicoletti, M.C.; Neto, A., "A Formal Approach to Consistency and Reuse of Links in World Wide Web Applications" *Formal Methods in Human-Computer Interaction*, Philippe Palanque & Fabio Paternò (eds.), Chapter 4, Springer-Verlag, p.75-92, 1997.

[11] Fortes, R.P.M., "Análise e Avaliação de Hiperdocumentos: uma abordagem baseada na Representação Estrutural" *Tese de Doutorado.* IFSC-USP, São Carlos-SP. 179p, 1996.

[12] Furnas, G.W. "Generalized fisheye views" In: ACM CHI'86 *Human Factors in Computing Systems*, Boston, April 1986. *Proceedings*. 1986. p.16-23.
[13] Garzotto, F.; Mainetti, L.; Paolini, P., "Hypermedia Design, Analysis and Evaluation Issues" *Comm. of the ACM* v. 38, n. 8, p.74-96, 1995.
[14] Halasz, F.G., "Reflections on Notecards: Seven Issues for the next generation of Hypermedia Systems" *Comm. of the ACM*, v.31, n.7, p.836-52, 1988.
[15] Halasz, F.; Schwartz, M. "The Dexter hypertext reference model" In: Moline,J.; Benigni,D.; Baronas,J., eds. *Proceedings of the NIST Hypertext Standardization Workshop* Gaithersburg, January 1990. National Institute of Standards and Technology Special Publication 500-178. Washington: U.S. Government Printing Office, 1990. p.95-133.
[16] Hardman, L. "Experiences in Authoring Hypermedia: Creating Better Presentations" In: Schuler, W.; Hannemann,J.; Streitz,N., eds. *Designing User Interfaces for Hypermedia*, Springer 1995. p.16-26.
[17] Hatzimanikatis, A.E.; Tsalidis, C.T.; Christodoulakis, D., "Measuring the Readability and Maintainability of Hyperdocuments" *Software Maintenance: Research and Practice*, v.7, p.77-90, 1995.
[18] Juran, J.M; Gryna Jr, F.M. *Quality Planning and Analysis: From Product Development Through Use*. New York USA, McGraw-Hill, 1970.
[19] Technical contents of ISO 639-2/1998 C*ode for the representation of names of languages:* Part 2: Alpha-3 code.
[20] Kollar, C.P.; Leavitt, J.R.R.; Mauldin, M. "Robot Exclusion Standard Revisted". http://www.kollar.com/robots.html
[21] Koster, M., "Guidelines for Robot Writes" http://info.webcrawler.com/mak/projects/robots/guidelines.html
[22] Koster, Martijn, "Robots in the Web: threat or treat?", ConneXions, Volume 9, No. 4, April 1995.
[23] Marshall, C.C.; Shipman, F.M., "Searching for the Missing Link: Discovering Implicit Structure in Spatial Hypertext" In: Hypertext'93, Seattle, Washington USA, November 1993. *Proceedings*. NewYork, ACM Press. p.217-30, 1993.
[24] Nielsen, J. "The Art of Navigating through Hypertext" *Comm. of the ACM*, v.33, n.3, p.296-310, March 1990.
[25] Nielsen, J., *Multimedia and Hypertext - The Internet and Beyond.* Academic Press, London, United Kingdom, 1995.
[26] Pinkerton, B. "Finding what people want: experiences with the WebCrawler", in: *Proc. of the 2nd International WWW Conference*, Chicago, USA, October 17–20, 1994.
[27] Salton, G.; Allan, J.; Buckley, C. "Automatic Structuring and Retrieval of Large Text Files" *Comm. of the ACM*, v. 37, .2, p.97-108, February 1994.
[28] Schwabe, D.; Rossi, G.; Barbosa, S.D.J. "Systematic Hypermedia Application Design with OOHDM" In: Hypertext'96, Washington DC, USA, March 1996. *Proceedings.* NewYork, ACM Press, 1996. p.116-28.
[29] Shibata, Y.; Katsumoto, M. "Dynamic Hypertext and Knowledge Agent Systems for Multimedia Information Networks" In: Hypertext'93, Seattle, Washington USA, November 1993. *Proceedings*. NewYork, ACM Press, 1993. p.82-93.
[30] Silva, A. S.; Veloso, E. A.; Golgher P. B.; Ribeiro-Neto B.; Ziviani, N.; Laender, A.H.F. "CobWeb - Um Coletor Automático de Documentos Web" *Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação, XXVI SEMISH*, Vol. 1, julho de 1999. p.233-247
[31] Utting, K.; Yankelovich, N. "Context and Orientation in Hypermedia Networks" *ACM Trans. on Information Systems*, v.7, n.1, p.58-84, January 1989.
[32] Young, L. "Linking Considered Harmful" In: Rizs,A.; Stritz,N. & André,J., eds. *Hypertext: Concepts, Systems and Applications* Cambridge, Great Britain, Cambridge University Press, 1990. p.238-49.