SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito: 19.06.2000

Assinatura:

Suporte Automático à Avaliação da Consistência Estrutural em WWW

Enzo Seraphim

Orientadora: Profa. Dra. Renata Pontin de Mattos Fortes

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Área: Ciências de Computação e Matemática Computacional.

USP – São Carlos
Junho de 2000

Aos meus pais Irineu e Malvina pelo
amor e apoio constante.

À minha irmã gêmea Zá pelo carinho
e compreensão.

Aos demais irmãos Leo, Cláudio,
Marcos, Soraia e Vica pelo incentivo.

Agradecimentos

Agradeço, primeiramente, a Deus pela capacidade e oportunidade de realizar esse trabalho, sempre procurando seguir seus princípios com muita fé e determinação, esperando alcançar suas expectativas.

À Prof^a Renata Pontin de Mattos Fortes, minha orientadora, pelo apoio e constante otimismo para que este trabalho fosse realizado.

Ao Prof. Edson dos Santos Moreira, pelo apoio e incentivo na etapa final do trabalho.

Ao Prof. Caetano Traina Junior pelo apoio e confiança depositada em mim.

Aos amigos Walter, Cláudio e Dani pelas discussões oferecidas, além de companherismo nas experiências do dia-a-dia.

Aos amigos Will, Cláudia, Gedson e Renato pela confiança e companherismo.

Aos amigos dos grupos de jovens Sinal de Esperança e Sementes do Futuro.

Aos demais amigos do ICMC-USP, que de uma maneira ou outra contribuíram com o estímulo para o prosseguimento deste trabalho.

Sumário

LISTA DE FIGURAS.....	III
LISTA DE TABELAS	IV
LISTA DE QUADROS.....	V
RESUMO	VI
ABSTRACT	VII
CAPÍTULO 1.....	1
1. INTRODUÇÃO	1
CAPÍTULO 2.....	6
2. CLASSIFICAÇÃO DE <i>LINKS</i> E PÁGINAS DE <i>WEB</i> SEGUNDO REUSO ESTRUTURAL.....	6
2.1 <i>Casos de links segundo o reuso de seus componentes</i>	6
2.1.1 Características da classificação dos <i>links</i>	9
2.2 <i>Casos de páginas segundo o reuso de seus links</i>	9
2.2.1 Características das páginas	12
2.2.1.1 Páginas com característica de navegação linear	12
2.2.1.2 Páginas com característica de navegação não linear.....	13
2.2.1.3 Páginas que não oferecem navegação (páginas folhas)	14
2.3 <i>Considerações Finais</i>	14
CAPÍTULO 3.....	15
3. FERRAMENTA DB-LIOS.....	15
3.1 <i>Histórico</i>	16
3.2 <i>Crawler</i>	17
3.3 <i>Classificação</i>	20
3.3.1 Classificação dos <i>links</i>	20
3.3.2 Classificação das páginas.....	23
3.4 <i>Base de Dados da ferramenta DB-LiOS</i>	24
3.4.1 Comunicação entre Base de Dados Local e Global	28
3.4.2 Transferência de dados para Base Global.....	28
3.5 <i>Interface da ferramenta DB-LiOS</i>	29
3.5.1 Interface sobre Base Local.....	29
3.5.2 Interface sobre Base Global	33
3.6 <i>Funcionalidades da Ferramenta DB-LiOS</i>	33
3.6.1 Funcionalidades sobre Base Local.....	34
3.6.2 Funcionalidades sobre Base Global	36
3.7 <i>Ferramenta DataImport</i>	36

3.8	<i>Considerações Finais</i>	37
CAPÍTULO 4		38
4.	EXPERIMENTOS COM DB-LIOS	38
4.1	<i>Descrição dos Experimentos Realizados</i>	38
4.2	<i>Considerações Finais</i>	45
CAPÍTULO 5		46
5.	FERRAMENTAS DE AVALIAÇÃO DE WEBSITES	46
5.1	<i>Ferramentas de Avaliação</i>	46
5.1.1	Ferramenta Linkbot Enterprise Server 2.0	46
5.1.2	Ferramenta WebAnalyser 1.1	48
5.1.3	Ferramenta LinkProver 2.1b	49
5.2	<i>Análise das Ferramentas</i>	50
5.3	<i>Considerações Finais</i>	52
CAPÍTULO 6		53
6.	CONCLUSÕES	53
6.1	<i>Conclusões Finais</i>	53
6.2	<i>Trabalhos futuros</i>	54
CAPÍTULO 7		55
7.	REFERÊNCIAS BIBLIOGRÁFICAS	55
APÊNDICES		59
	APÊNDICE A - BASE DE DADOS EM PARADOX DO DB-LIOS	59
	APÊNDICE B - BASE DE DADOS EM POSTGRESQL DO DB-LIOS	63
	APÊNDICE C - CÓDIGOS DE STATUS DE HTTP	66

Lista de Figuras

FIGURA 1 - CASOS DE REUSO DE <i>LINKS</i> EM WEBSITE.....	8
FIGURA 2 - MÓDULOS FUNCIONAIS DO DB-LIOS.....	15
FIGURA 3 - TELA DA FERRAMENTA DB-LIOS	16
FIGURA 4 - BASE DE DADOS RELACIONAL IMPLEMENTADA EM DB-LIOS.	26
FIGURA 5 - SISTEMA DE COMUNICAÇÃO DA FERRAMENTA DB-LIOS.	28
FIGURA 6 - JANELA DE USUÁRIO DA FERRAMENTA DB-LIOS.....	29
FIGURA 7 - DADOS SOBRE <i>SITES</i>	30
FIGURA 8 - DADOS SOBRE A(S) VERSÃO(ÕES) DO <i>SITE</i>	30
FIGURA 9 - DADOS ESTATÍSTICOS SOBRE A VERSÃO DO <i>SITE</i>	31
FIGURA 10 - GRÁFICO DA CLASSIFICAÇÃO DOS <i>LINKS</i>	31
FIGURA 11 - DADOS DAS PÁGINAS.	32
FIGURA 12 - DADOS DOS <i>LINKS</i>	32
FIGURA 13 - CLASSIFICAÇÃO DOS <i>LINKS</i>	33
FIGURA 14 - MENU E BARRA DE FERRAMENTAS DO DB-LIOS	33
FIGURA 15 - TELA PARA ADICIONAR UM NOVO SITE OU UMA NOVA VERSÃO DO <i>SITE</i>	35
FIGURA 16 - MENU <i>FILE</i> DA FERRAMENTA DB-LIOS.....	35
FIGURA 17 - MENU <i>OPERATION</i> DA FERRAMENTA DB-LIOS	36
FIGURA 18 - FERRAMENTA DATAIMPORT	36
FIGURA 19 - PERCENTUAIS MÉDIOS DOS CASOS DE REUSO DE <i>LINKS</i> DOS SITES OBTIDOS PELO DB-LIOS.....	42
FIGURA 20 - TELA DO LINKBOT ENTERPRISE SERVER 2.0.....	47
FIGURA 21 - TELA DA FERRAMENTA WEBANALYSER 1.1	49
FIGURA 22 - TELA DA FERRAMENTA LINKPROVER 2.1B	50

Lista de Tabelas

TABELA 1 - TABELA DE CLASSIFICAÇÃO DE PÁGINAS.....	10
TABELA 2 - RESULTADOS SOBRE OS SITES OBTIDOS PELO DB-LIOS.	39
TABELA 3 - RESULTADOS SOBRE OS <i>LINKS</i> DOS SITES OBTIDOS PELO DB-LIOS.	40
TABELA 4 - RESULTADOS SOBRE AS CLASSIFICAÇÕES DOS <i>LINKS</i> DOS SITES OBTIDOS PELO DB-LIOS.....	41
TABELA 5 - TOTALIZAÇÃO DOS CASOS DE PÁGINAS OBTIDOS PELO DB-LIOS	44
TABELA 6 - ANÁLISE DAS FUNCIONALIDADES ENTRE AS FERRAMENTAS DE AVALIAÇÃO	51
TABELA 7 - ANÁLISE DOS EXPERIMENTOS ENTRE AS FERRAMENTAS DE AVALIAÇÃO	51
TABELA 8 - TABELA DE USUÁRIOS	59
TABELA 9 - DESCRIÇÃO DA TABELA DE SITES	59
TABELA 10 - DESCRIÇÃO DA TABELA DE VERSÃO DO SITE.....	60
TABELA 11 - DESCRIÇÃO DA TABELA DE URL.....	60
TABELA 12 - DESCRIÇÃO DA TABELA DE CÓDIGO DE STATUS HTTP	60
TABELA 13 - TABELA DE PÁGINAS VISITADAS NO SITE	61
TABELA 14 - TABELA DE ÂNCORAS DOS <i>LINKS</i>	61
TABELA 15 - TABELA DE <i>LINKS</i> DAS PÁGINAS DO SITE.....	61
TABELA 16 - TABELA DE CLASSIFICAÇÃO DE <i>LINKS</i>	62
TABELA 17 - TABELA DE CÓDIGOS DE STATUS HTTP	66

Lista de Quadros

QUADRO 1 - CLASSIFICAÇÃO DE <i>LINKS</i> NA BASE 2.....	10
QUADRO 2 - ALGORITMO DE CLASSIFICAÇÃO DO LINK0	21
QUADRO 3 - ALGORITMO DE CLASSIFICAÇÃO DO LINK1	21
QUADRO 4- ALGORITMO DE CLASSIFICAÇÃO DO LINK2	21
QUADRO 5 - ALGORITMO DE CLASSIFICAÇÃO DO LINK3	22
QUADRO 6 - ALGORITMO DE CLASSIFICAÇÃO DO LINK4	22
QUADRO 7 - ALGORITMO DE CLASSIFICAÇÃO DO LINK5	22
QUADRO 8 - ALGORITMO DE CLASSIFICAÇÃO DO LINK6	23
QUADRO 9 - ALGORITMO DE CLASSIFICAÇÃO DO LINK7	23
QUADRO 10 - ALGORITMO DE CLASSIFICAÇÃO DE PÁGINAS	24

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*¹, para que o usuário se depare com uma inconsistência e se sinta “perdido”. Um procedimento comum durante o desenvolvimento de um *site*² 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*.

¹ *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.

² *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.

Abstract

The dynamics and flexibility of *Websites* authoring, on the one hand, popularize the Internet usage increasingly. 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*.

Capítulo 1

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” [Conklin 1987] e [Halasz 1988] 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.

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 `</a>`, 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 [Berners-Lee et al. 1994].

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 [Hatzimanikatis et al 1995]. Neste contexto, uma análise estrutural com base em uma classificação dos *links* de um hipertexto, segundo a reusabilidade de seus componentes, foi definida [Fortes 1996]. A partir dessa classificação de *links*, métricas baseadas em reuso de *links* são formuladas [Fortes & Nicoletti 1997], das quais valores podem ser coletados periodicamente de forma a auxiliar na administração do processo de manutenção de um *website*.

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 [Juran & Gryna Jr 1970].

A primeiras soluções propostas aos problemas inerentes à utilização de hipertextos envolviam investimentos em análise de textos e melhorias dos sistemas de navegação disponíveis [Botafogo & Shneiderman 1992]. Entre os trabalhos mais representativos na área de análise de textos encontram-se os desenvolvidos por Salton et al. [Salton et al 1994], 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 [Hardman 1995]. Exemplos dessa abordagem incluem: diagramas gráficos de visão-geral gerados automaticamente e diagramas gráficos de visão-geral de contextos específicos gerados manualmente [Nielsen 1990]; visões de *Web* do Intermedia (*Web Vision*) [Utting & Yankelovich 1989]; visões de olho-de-peixe (*fisheye*) [Furnas 1986]; agentes de conhecimento [Shibata & Katsumoto 1993]. 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 [Bernstein 1990] 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 [Halasz 1988] 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 [Hatzimanikatis et al 1995].

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. [Schwabe et al 1996] 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*³ possuem as características de sistema hipertexto, mas com particularidades. Seguindo o modelo de referência de Dexter [Halasz & Schwartz 1990] 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 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

³ exemplos de *browsers* para *Web* são Netscape, Explorer

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 [Brown 1990]. 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 [Marshall & Shipman 1993].
- 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 [Nielsen 1995].
- 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 [Young 1990].

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.

Esta dissertação 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* [Fortes & Nicoletti 1997]. 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 [Garzotto et al 1995], compõem a abordagem de cinco dimensões que ser considerada para se avaliar um hiperdocumento: conteúdo, estrutura, apresentação, dinâmica e interação.

Esta dissertação está organizada da seguinte forma: o Capítulo 2 descreve a classificação de *links* e de páginas de *Web* segundo reuso estrutural os quais fundamentam a base de *links* desenvolvida em DB-LiOS. No Capítulo 3 encontram-se descritas as principais características funcionais e estruturais da arquitetura desenvolvida na ferramenta DB-LiOS. No Capítulo 4 são apresentados os experimentos realizados e os resultados obtidos pela utilização de DB-LiOS. O Capítulo 5 apresenta uma revisão de algumas ferramentas comerciais da atualidade que se propõem a avaliar *Websites*. Finalmente, no Capítulo 6 são apresentadas as próximas pesquisas, bem como as conclusões deste trabalho.

2. Classificação de *links* e páginas de *Web* segundo reuso estrutural

A principal característica de sistemas hipermídia é a possibilidade de leitura não linear de um documento proporcionada pelo uso de *links*. Em quase toda página de um *site* existem *links* que permitem ao usuário navegar dentro do *site* ou mesmo para fora dele. Na *Web*, a partir dos *links*, através dos quais se navega dentro do *site*, é possível determinar a estrutura de páginas HTML que formam o *site*. Sobre essa estrutura que os *webmasters*, que gerenciam o site, possuem controle para edição.

A composição página-origem, âncora e página-destino formam o *link* estático básico na *Web*. Este *link* está definido em uma página denominada página-origem que apresenta uma âncora, que pode ser um texto ou uma imagem, e que direciona para uma outra página ou outra posição da mesma página (denominada página-destino). Em todo *site* reutilizamos partes desta composição do *link*. Esta reutilização pode ser feita na página-origem quando temos mais de um *link* na mesma página, ou ser feita quando repetimos a mesma âncora em diversas páginas ou na própria página, ou ainda quando vários *links* direcionam para uma mesma página-destino.

Neste capítulo são descritas as abordagens de reuso de *links* que fundamentam os mecanismos implementados na ferramenta desenvolvida: a classificação dos *links* de [Fortes 1996] bem como a classificação de páginas segundo os casos de *links*.

2.1 Casos de *links* segundo o reuso de seus componentes

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 [Fortes 1996]:

Link0 - *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*.

Link1 - *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*.

Link2 - *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*.

Link3 - *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*.

Link4 - *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*.

Link5 - *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*.

Link6 - *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*.

Link7 - *links* que estão na mesma página-origem com a mesma âncora e direcionam para a mesma página-destino.

A Figura 1 esquematiza, de forma resumida, esses oito casos de reuso de componentes de *links* (Link0 a Link7) de um *website*. A Figura 1 apresenta também, para cada um dos casos, o número de *links* possíveis de serem encontrados em cada página-origem do *site*.

Figura 1 - Casos de reuso de links em website.

Obs: Nesta figura, 0 significa diferente (ou seja, reuso = falso) e 1 significa igualdade.

2.1.1 Características da classificação dos *links*

Segundo os casos de reuso de *links* propostos em [Fortes & Nicoletti 1997] e [Fortes et al 1997], 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:

Link0 - a página que possui este caso apresenta somente este único *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*;

Link1, Link2 e Link3 - as páginas que possuem esses casos apresentam um ou mais *links*, pois não há reuso de página-origem.

Link4, Link5, Link6 e Link7 - 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 na Figura 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 proposta por Fortes (Fortes, 1996).

É importante lembrar que um *link* pode estar classificado em um ou mais casos de reusabilidade (Fortes, 1996).

2.2 Casos de páginas segundo o reuso de seus *links*

A partir da classificação de *links* estudada, identificamos a classificação de páginas segundo os casos de reuso de *links* que nela apresente. Esta classificação das páginas permite analisar como os casos de *links* estão distribuídos em cada página do *site*.

Para se estabelecer a quantidade de casos de classificação de páginas, pensemos inicialmente que cada caso de *link* pode existir ou não em uma página. Como são oito, os casos de *links*, conclui-se que existem 2^8 casos de classificação de páginas, ou seja, 256 casos. No entanto, segundo a restrição de quantidade mínima de *links* (quando uma página apresenta o link0 ela não pode apresentar nenhum outro *link*) temos que diminuir assim o total de casos de classificação de páginas para $2^7 + 1$, ou seja, 129 casos. E conforme mostra o Quadro 1, para cada caso de *link* foi definido um valor (em base 2).

Quadro 1 - Classificação de *links* na base 2.

Link0 $\Rightarrow$ 128	(2 ⁷)	Link1 $\Rightarrow$ 64	(2 ⁶)	Link2 $\Rightarrow$ 32	(2 ⁵)	Link3 $\Rightarrow$ 16	(2 ⁴)
Link4 $\Rightarrow$ 8	(2 ³)	Link5 $\Rightarrow$ 4	(2 ²)	Link6 $\Rightarrow$ 2	(2 ¹)	Link7 $\Rightarrow$ 1	(2 ⁰)

A Tabela 1 mostra as classificações de páginas com base nos casos de *links*.

Tabela 1 - Tabela de Classificação de Páginas.

Classificação	Link0 <i>bin</i> 128	Link1 <i>Bin</i> 64	Link2 <i>bin</i> 32	Link3 <i>bin</i> 16	Link4 <i>bin</i> 8	Link5 <i>bin</i> 4	Link6 <i>bin</i> 2	Link7 <i>bin</i> 1
Page 0	0	0	0	0	0	0	0	0
Page 1	0	0	0	0	0	0	0	1
Page 2	0	0	0	0	0	0	1	0
Page 3	0	0	0	0	0	0	1	1
Page 4	0	0	0	0	0	1	0	0
Page 5	0	0	0	0	0	1	0	1
Page 6	0	0	0	0	0	1	1	0
Page 7	0	0	0	0	0	1	1	1
Page 8	0	0	0	0	1	0	0	0
Page 9	0	0	0	0	1	0	0	1
Page 10	0	0	0	0	1	0	1	0
Page 11	0	0	0	0	1	0	1	1
Page 12	0	0	0	0	1	1	0	0
Page 13	0	0	0	0	1	1	0	1
Page 14	0	0	0	0	1	1	1	0
Page 15	0	0	0	0	1	1	1	1
Page 16	0	0	0	1	0	0	0	0
Page 17	0	0	0	1	0	0	0	1
Page 18	0	0	0	1	0	0	1	0
Page 19	0	0	0	1	0	0	1	1
Page 20	0	0	0	1	0	1	0	0
Page 21	0	0	0	1	0	1	0	1
Page 22	0	0	0	1	0	1	1	0
Page 23	0	0	0	1	0	1	1	1
Page 24	0	0	0	1	1	0	0	0
Page 25	0	0	0	1	1	0	0	1
Page 26	0	0	0	1	1	0	1	0
Page 27	0	0	0	1	1	0	1	1
Page 28	0	0	0	1	1	1	0	0
Page 29	0	0	0	1	1	1	0	1
Page 30	0	0	0	1	1	1	1	0
Page 31	0	0	0	1	1	1	1	1
Page 32	0	0	1	0	0	0	0	0
Page 33	0	0	1	0	0	0	0	1
Page 34	0	0	1	0	0	0	1	0
Page 35	0	0	1	0	0	0	1	1
Page 36	0	0	1	0	0	1	0	0
Page 37	0	0	1	0	0	1	0	1
Page 38	0	0	1	0	0	1	1	0
Page 39	0	0	1	0	0	1	1	1
Page 40	0	0	1	0	1	0	0	0
Page 41	0	0	1	0	1	0	0	1
Page 42	0	0	1	0	1	0	1	0
Page 43	0	0	1	0	1	0	1	1
Page 44	0	0	1	0	1	1	0	0
Page 45	0	0	1	0	1	1	0	1
Page 46	0	0	1	0	1	1	1	0
Page 47	0	0	1	0	1	1	1	1
Page 48	0	0	1	1	0	0	0	0
Page 49	0	0	1	1	0	0	0	1
Page 50	0	0	1	1	0	0	1	0
Page 51	0	0	1	1	0	0	1	1
Page 52	0	0	1	1	0	1	0	0
Page 53	0	0	1	1	0	1	0	1
Page 54	0	0	1	1	0	1	1	0
Page 55	0	0	1	1	0	1	1	1
Page 56	0	0	1	1	1	0	0	0
Page 57	0	0	1	1	1	0	0	1
Page 58	0	0	1	1	1	0	1	0
Page 59	0	0	1	1	1	0	1	1

Page 60	0	0	1	1	1	1	0	0
Page 61	0	0	1	1	1	1	0	1
Page 62	0	0	1	1	1	1	1	0
Page 63	0	0	1	1	1	1	1	1
Page 64	0	1	0	0	0	0	0	0
Page 65	0	1	0	0	0	0	0	1
Page 66	0	1	0	0	0	0	1	0
Page 67	0	1	0	0	0	0	1	1
Page 68	0	1	0	0	0	1	0	0
Page 69	0	1	0	0	0	1	0	1
Page 70	0	1	0	0	0	1	1	0
Page 71	0	1	0	0	0	1	1	1
Page 72	0	1	0	0	1	0	0	0
Page 73	0	1	0	0	1	0	0	1
Page 74	0	1	0	0	1	0	1	0
Page 75	0	1	0	0	1	0	1	1
Page 76	0	1	0	0	1	1	0	0
Page 77	0	1	0	0	1	1	0	1
Page 78	0	1	0	0	1	1	1	0
Page 79	0	1	0	0	1	1	1	1
Page 80	0	1	0	1	0	0	0	0
Page 81	0	1	0	1	0	0	0	1
Page 82	0	1	0	1	0	0	1	0
Page 83	0	1	0	1	0	0	1	1
Page 84	0	1	0	1	0	1	0	0
Page 85	0	1	0	1	0	1	0	1
Page 86	0	1	0	1	0	1	1	0
Page 87	0	1	0	1	0	1	1	1
Page 88	0	1	0	1	1	0	0	0
Page 89	0	1	0	1	1	0	0	1
Page 90	0	1	0	1	1	0	1	0
Page 91	0	1	0	1	1	0	1	1
Page 92	0	1	0	1	1	1	0	0
Page 93	0	1	0	1	1	1	0	1
Page 94	0	1	0	1	1	1	1	0
Page 95	0	1	0	1	1	1	1	1
Page 96	0	1	1	0	0	0	0	0
Page 97	0	1	1	0	0	0	0	1
Page 98	0	1	1	0	0	0	1	0
Page 99	0	1	1	0	0	0	1	1
Page 100	0	1	1	0	0	1	0	0
Page 101	0	1	1	0	0	1	0	1
Page 102	0	1	1	0	0	1	1	0
Page 103	0	1	1	0	0	1	1	1
Page 104	0	1	1	0	1	0	0	0
Page 105	0	1	1	0	1	0	0	1
Page 106	0	1	1	0	1	0	1	0
Page 107	0	1	1	0	1	0	1	1
Page 108	0	1	1	0	1	1	0	0
Page 109	0	1	1	0	1	1	0	1
Page 110	0	1	1	0	1	1	1	0
Page 111	0	1	1	0	1	1	1	1
Page 112	0	1	1	1	0	0	0	0
Page 113	0	1	1	1	0	0	0	1
Page 114	0	1	1	1	0	0	1	0
Page 115	0	1	1	1	0	0	1	1
Page 116	0	1	1	1	0	1	0	0
Page 117	0	1	1	1	0	1	0	1
Page 118	0	1	1	1	0	1	1	0
Page 119	0	1	1	1	0	1	1	1
Page 120	0	1	1	1	1	0	0	0
Page 121	0	1	1	1	1	0	0	1
Page 122	0	1	1	1	1	0	1	0
Page 123	0	1	1	1	1	0	1	1
Page 124	0	1	1	1	1	1	0	0
Page 125	0	1	1	1	1	1	0	1
Page 126	0	1	1	1	1	1	1	0
Page 127	0	1	1	1	1	1	1	1
Page 128	1	0	0	0	0	0	0	0

0 - Não existe / 1 - Existe

A classificação de cada página foi proposta com base na soma dos valores dos casos de *links* que existem na página. Independente do número de vezes que a classificação de *link* aparece na página, ou seja, somente é somada uma única vez o valor correspondente.

Por exemplo: quando referirmos à página que é classificada 99 (01100011_2), estamos nos referindo àquela página cuja combinação de *links* seja a soma $64 + 32 + 2 + 1$, ou seja, $\text{link1} + \text{link2} + \text{link6} + \text{link7}$, independentemente de quantas vezes aparecem o link1 , link2 , link6 ou link7 na página.

2.2.1 Características das páginas

A grande quantidade de classes de páginas dificulta o estudo sobre o que significa cada caso. No entanto, observamos um padrão na classificação de páginas quanto à *navegação linear e não linear*.

Partimos da idéia de que quando uma página oferece navegação estritamente linear, nela só existe um *link*, que mostra o prosseguimento da seqüência linear. Quando a página possui mais de um *link*, é considerada como página de navegação não exclusivamente linear. As páginas que não possuem *links* são chamadas de *páginas folhas*.

2.2.1.1 Páginas com característica de navegação linear

Quando navegamos pelas páginas da *Web*, a cada instante temos que tomar decisões frente à multiplicidade de escolhas que nos é oferecida. As informações no hipertexto não são mais dispostas linearmente (seqüencialmente) como estamos acostumados a ver nos documentos impressos.

Essa nova forma de disponibilização é atrativa, pois possibilita que o leitor exerça uma interação que lhe auxilie na busca de suas informações. No entanto há circunstâncias em que o autor deseja limitar a liberdade navegacional do usuário oferecendo uma navegação linear. Para isso, o autor oferece apenas um *link* no documento HTML. Assim, procuramos estudar quais as classificações de páginas oferecem apenas um *link*.

Como visto anteriormente, quando a página apresenta o link0 só pode existir nela esse único *link*. A única classificação de página que tem o link0 é a **Page 128** (10000000_2). Portanto, do ponto de vista estrutural a Page 128 é uma possível página de navegação linear. Vale ressaltar que não podemos afirmar categoricamente que esta é uma *navegação linear* pois esta análise requereria uma investigação sobre o conteúdo do documento.

Conforme Figura 1, quando uma página existir o caso de classificação de link4, link5, link6 ou link7, essa página possui dois ou mais *links*. Essas classificações, que possuem esses casos, não possuem navegação linear.

Quanto à análise das páginas em que aparecem os casos link1, link2 e link3, imagine que uma página contenha dois *links*: mesmo que os *links* sejam classificados como um link1 e o outro como link2. Dessa forma, eles também seriam classificados em um dos casos de reuso de página-origem (link4 ou link5 ou link6 ou link7). O aparecimento de qualquer classificação dos casos de reuso de página-origem determina que a página tem dois ou mais *links*.

Portanto, para que uma página apresente apenas um *link*, nela só podem existir *links* classificados entre as combinações de link1, link2 e link3. A seguir, estão essas combinações:

- **Page 64** (01000000_2) onde na página só podem existir *links* classificados como link1;
- **Page 32** (00100000_2) onde na página só podem existir *links* classificados como link2;
- **Page 16** (00010000_2) onde na página só podem existir *links* classificados como link3;
- **Page 96** (01100000_2) onde na página só podem existir *links* classificados como link1 e link2;
- **Page 80** (01010000_2) onde na página só podem existir *links* classificados como link1 e link3;
- **Page 112** (01110000_2) onde na página só podem existir *links* classificados como link1, link2 e link3;
- **Page 48** (00110000_2) onde na página só podem existir *links* classificados como link2 e link3;

Conclui-se que as classificações de páginas Page 16, Page 32, Page 48, Page 64, Page 80, Page 96, Page 112 e Page 128 são páginas de *navegação linear*.

2.2.1.2 *Páginas com característica de navegação não linear*

Quando uma página oferece múltiplas opções de escolha para navegação, a classificamos como sendo uma página que permite *navegação não linear*. Portanto essa página deve conter dois ou mais *links*.

Para que uma página apresente dois ou mais *links* ela deve conter *links* classificados em um dos casos de reuso de página-origem, ou seja, link4 ou link5 ou link6 ou link7.

Todas as páginas, excluindo-se as que são classificadas como páginas de navegação linear e as páginas que não oferecem navegação (Page 0), são classificadas como sendo páginas de navegação não linear.

Dentre todas as classificações, destacamos a classificação Page 127 (01111111₂) por conter o número máximo de casos de *links* em uma única página, ou seja, Link1 + Link2 + Link3 + Link4 + Link5 + Link6 + Link7.

2.2.1.3 *Páginas que não oferecem navegação (páginas folhas)*

A principal característica de sistemas de hiperdocumentos é a navegação. No entanto, uma página pode não conter nenhum *link*. Essas páginas são conhecidas também como sendo páginas folhas, pois não permitem mais navegação ao usuário.

As páginas classificadas como sendo Page 0, são as que não contém *links*.

2.3 Considerações Finais

Neste capítulo foi descrita a classificação de *links* proposta por Fortes [Fortes 1996]. A proposta de classificação de páginas, segundo a existência dos casos de classificação de *links*, é também apresentada. Através da classificação de páginas pode-se determinar se uma página oferece: navegação linear, navegação não linear ou não oferece navegação.

As classificações de *links* e de páginas proporcionam uma análise quanto a reusabilidade de componentes de *links* na *Web*. O próximo capítulo apresenta a ferramenta DB-LiOS que foi desenvolvida para auxiliar esta análise de reusabilidade estrutural de documentos da *Web*.

3. Ferramenta 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, uma ferramenta de auxílio automático denominada DB-LiOS (*DataBase - Link Oriented System*) foi desenvolvido, por meio de uma avaliação do reuso de *links* de um *website*, com base na classificação de *links* de Fortes [Fortes & Nicoletti 1997] [Fortes et al 1997] [Fortes 1996]. De fato, este auxílio visa não somente atender o enorme volume de informações a serem mantidas nos *Websites*, como também a complexidade da atividade de manutenção uma vez que os *links* de um *website* podem estar embutidos em qualquer local inserido nas diversas páginas do hiperdocumento.

Como pode ser observado na Figura 2, existem 3 módulos funcionais no DB-LiOS: a) o crawler que extrai os componentes estruturais do site; b) o classificador de links nos 8 casos propostos por Fortes e c) o classificador de páginas nos 129 casos propostos nesse trabalho.

Figura 2 - Módulos Funcionais do DB-LiOS.

De forma a explicar a ferramenta DB-LiOS o capítulo encontra-se organizado da seguinte forma: inicialmente um histórico sobre as evoluções da ferramenta. Nas seções posteriores são descritas os módulos funcionais da ferramenta (*crawler*, classificador de *links* e de páginas), a base de dados, sua interface e suas funcionalidades. Finalmente são apresentadas algumas considerações finais sobre a ferramenta DB-LiOS.

3.1 Histórico

A ferramenta DB-LiOS (Figura 3) foi desenvolvida inicialmente para extrair e classificar os *links* nos casos propostos por Fortes [Fortes 1996] utilizando-se para isso ambiente de programação Delphi [Seraphim & Fortes 1999] [Blue 1997] na plataforma WindowsTM

Figura 3 - Tela da ferramenta DB-LiOS

Com a adição da classificação de páginas, descrita no capítulo anterior, a ferramenta foi modificada para classificar também as páginas que contém os *links*. Além disso, observou-se a necessidade do tratamento de dois requisitos para melhor funcionalidade da ferramenta: o

controle de versões e o compartilhamento das informações. Esses requisitos foram incorporados à ferramenta e são descritos a seguir.

Controle de versão

É muito comum que nos *sites* ocorram muitas mudanças em suas páginas. Pela própria natureza de novidade tecnológica, os hiperdocumentos tendem a estar em constante evolução. Mas para que não ocorram inconsistências entre as páginas do *site*, as mudanças necessitam ser controladas através de um controle de versões. Pesquisas científicas sugerem meta *tags* especiais para incluir controle de versão em páginas HTML [Vitali & Durand 1995].

A ferramenta DB-LiOS considera os frequentes processos de evolução de um *site*, que originam mudanças nas suas páginas e seus *links*. No DB-LiOS, essas mudanças podem ser gerenciadas através de um controle de versão do *site*. Para assegurar esse controle de versões, foi definido no modelo relacional da base de dados de DB-LiOS uma tabela de controle de versões como mostra a Figura 4.

Compartilhamento de informações

Durante o período em que a ferramenta foi submetida à extração/classificação de vários *sites*, pudemos observar-se a necessidade de que cada versão possuísse um sumário de cada classificação de *links* e páginas, a que fosse submetido. Mais importante, que essas informações pudessem ser compartilhadas entre todos os usuários que utilizassem a ferramenta DB-LiOS. Para assegurar compartilhamento de informações, foi decidido criar 2 bases de dados: uma Local em cada instalação da ferramenta, a qual conteria todas as informações de extração/classificação de cada versão do *site* e outra Global, a qual conteria as sumarizações dessas extrações/classificações.

3.2 Crawler

Crawlers também são conhecidos como *spiders*, *Web wanderers*, *Web worms* e *robots* [Koster 1995]. O *crawler* [Cho et al 1998] é um programa autônomo que navega por todas páginas do *site* através de seus *links*. Os *crawlers* são utilizados para diversos fins [Koster 1995], por exemplo, na coleta de palavras para formação de catálogos de *Search Engine* [Pinkerton 1994] (como vistos em AltaVista, Yahoo, WebCrawler e Excite) e na coleta de todos os documentos que compõem o *site* para criar cópias espelhos do *site* [Silva et al 1999]. Neste trabalho, a função

do *crawler* em DB-LiOS é extrair as informações referentes aos componentes do *links*: página-origem, âncora e página-destino.

À medida que o *crawler* percorre as páginas de um *site*, são geradas instâncias dos *links* na Base de Dados Local, compostas pelos dados estruturais de um *site* como a identificação de seus *links* e respectivas páginas. Para o desenvolvimento do *crawler* no 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 [Kollar 1999]. 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 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 que um servidor HTTP não sobrecarregue. Em [Koster 1999] o intervalo sugerido é de pelo menos 1 minuto. No DB-LiOS o usuário pode definir 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 pode definir o tempo máximo de espera a uma requisição. Quando o tempo se esgota a URL é adicionada novamente no final da lista.

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 pode definir a quantidade máxima de tentativas.

e) escalonamento das URLs – no DB-LiOS foi implementado um escalonador baseado na política FIFO, ou seja, as primeiras URLs extraídas das páginas são as primeiras a serem requisitadas, observado o tempo de espera entre requisições item *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 [Fielding 1999] 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. Embora a ferramenta classifique todos os códigos HTTP retornados pelo servidor de *Web*, ela somente trata cinco tipos de páginas:

1. Boas → as páginas que foram lidas totalmente, ou seja o código HTTP igual 200;
2. Não lidas → as páginas que estão na fila de espera;
3. Expirou tempo de espera → as páginas que excederam o número máximo de vezes de releitura. Existe ainda um tempo de máximo de espera para a leitura da página, quando expira o tempo de espera, a página é colocada no final da fila para tentar reler seu conteúdo. Páginas classificadas como expirou timeout recebem código -1 (menos um);
4. Código HTTP Desconhecido → as páginas que o servidor de HTTP retorna com um código de HTTP desconhecido. As páginas classificadas como código HTTP desconhecido recebem código -2 (menos dois);
5. Mortas → as páginas que o servidor de HTTP retornou o código de status igual 404;

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, nesse 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.

O processo de extração pode ser cancelado manualmente pelo usuário ocasionando assim a não leitura de algumas páginas. Para que sejam lidas foi implementado o processo de releitura que consiste em colocar na lista de espera para leitura as páginas classificadas como: não lidas, expirou *timeout*, código http desconhecido.

3.3 Classificação

Executado o processo de extração dos *links*, inicia-se o processo de classificação dos *links* e de páginas descritos no capítulo anterior. Esse 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 muito dispendioso, optou-se armazenar os resultados de classificação na base de dados.

Uma vez iniciado o processo de classificação não pode ser cancelado pelo usuário, pois o cancelamento implicaria em deixar incorreta a classificação de *links* e de páginas. Os algoritmos de classificação foram implementados na própria ferramenta, evitando assim a realização de seleções avançadas não suportadas por alguns SGBDs.

A classificação de *links* verifica a repetição de página-origem, âncora e página-destino de um *link* de uma versão do *site* enquanto a classificação de páginas verifica a existência na página dos casos de classificação de *links*.

3.3.1 Classificação dos *links*

Após o término do processo de extração de uma versão do *site*, cada *link* extraído, composto por página-origem, âncora e página-destino, forma um registro na tabela de *links*. O processo de classificação dos *links* consiste em classificar os *links* nos 8 casos o propostos por Fortes [Fortes 1996].

Esta etapa de classificação de *links* verifica a repetição de página-origem, âncora e página-destino de cada *link* de uma 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 na tabela de *link*.

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.

Nos quadros apresentados a seguir, Quadro 2 a Quadro 9, são descritos os algoritmos de cada caso de classificação de *links*.

Quadro 2 - Algoritmo de Classificação do Link0

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$
O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se o_1 não se repete dentro do conjunto V ;
 - ii) Se d_1 não se repete dentro do conjunto V ;
 - iii) Se a_1 não se repete dentro do conjunto V ;Então L_1 é classificado como **link0**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 3 - Algoritmo de Classificação do Link1

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$
O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se d_1 repete dentro do conjunto V ;
 - ii) Se a_1 não se repete dentro do conjunto V ;
 - iii) Se o_1 não se repete dentro do conjunto V ;Então L_1 é classificado como **link1**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 4- Algoritmo de Classificação do Link2

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$
O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se a_1 repete dentro do conjunto V ;
 - ii) Se o_1 não se repete dentro do conjunto V ;
 - iii) Se d_1 não se repete dentro do conjunto V ;Então L_1 é classificado como **link2**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 5 - Algoritmo de Classificação do Link3

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$

O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se a_1 repete dentro do conjunto V ;
 - ii) Se d_1 repete dentro do conjunto V ;
 - iii) Se o_1 não se repete dentro do conjunto V ;
Então L_1 é classificado como **link3**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 6 - Algoritmo de Classificação do Link4

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$

O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se o_1 repete dentro do conjunto V ;
 - ii) Se d_1 não repete dentro do conjunto V ;
 - iii) Se a_1 não se repete dentro do conjunto V ;
Então L_1 é classificado como **link4**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 7 - Algoritmo de Classificação do Link5

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$

O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se o_1 repete dentro do conjunto V ;
 - ii) Se d_1 repete dentro do conjunto V ;
 - iii) Se a_1 não se repete dentro do conjunto V ;
Então L_1 é classificado como **link5**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 8 - Algoritmo de Classificação do Link6

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$

O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se o_1 repete dentro do conjunto V ;
 - ii) Se a_1 repete dentro do conjunto V ;
 - iii) Se d_1 não se repete dentro do conjunto V ;
Então L_1 é classificado como **link6**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

Quadro 9 - Algoritmo de Classificação do Link7

Dada uma versão do site $\underline{V}$ que é um conjunto de links L . Onde $V = \{L_1, L_2, L_3, \dots, L_n\}$

O link L_1 é formado por página origem, âncora e página destino. Onde $L_1 = (o_1, a_1, d_1)$.

- 1) Escolhe o primeiro link L_1 ;
- 2)
 - i) Se o_1 repete dentro do conjunto V ;
 - ii) Se d_1 repete dentro do conjunto V ;
 - iii) Se a_1 repete dentro do conjunto V ;
Então L_1 é classificado como **link7**;
- 3) Escolhe o próximo link do conjunto V e repete passo 2 até que todos os links do conjunto V sejam verificados.

3.3.2 Classificação das páginas

Realizado o processo de classificação dos *links* de uma versão do *site*, todos os *links* estão classificados em um ou mais casos de classificação de *links*. Dessa forma pode ser iniciado o processo de classificação das páginas, que verifica quais casos de *links* cada página possui. As páginas podem ser classificadas em 129 casos, conforme mencionando no capítulo anterior.

No Quadro 10 , a seguir, é feita uma descrição do algoritmo de classificação das páginas.

Quadro 10 - Algoritmo de classificação de páginas

Dada uma versão do site V que é um conjunto de links classificados L_c .

Onde $V = \{Lc_1, Lc_2, Lc_3, \dots, Lc_n\}$

O link classificado Lc_1 é formado por página origem e a classificação do link.

Onde $Lc_1 = (o_1, c_1)$.

O conjunto V está ordenado por página de origem.

- 1) Escolhe o primeiro link Lc_1 ;
- 2) Inicializa CasoPágina = 0;
- 3) Se c_1 é igual 0; então CasoPágina = CasoPágina + 128;
- 4) Se c_1 é igual 1; então CasoPágina = CasoPágina + 64;
- 5) Se c_1 é igual 2; então CasoPágina = CasoPágina + 32;
- 6) Se c_1 é igual 3; então CasoPágina = CasoPágina + 16;
- 7) Se c_1 é igual 4; então CasoPágina = CasoPágina + 8;
- 8) Se c_1 é igual 5; então CasoPágina = CasoPágina + 4;
- 9) Se c_1 é igual 6; então CasoPágina = CasoPágina + 2;
- 10) Se c_1 é igual 7; então CasoPágina = CasoPágina + 1;
- 11) Escolhe o próximo link classificado do conjunto V ;
- 12) Se o_1 não mudou de página,
Então repete passo 3;
- 13) Se o_1 mudou o valor,
Então a página o_1 é classificada com o valor de CasoPágina;
Repete passo 2 até que todos os links classificados do conjunto V sejam verificados.

As páginas origem que não contém links, ou seja, que não contém elementos no conjunto V , são classificadas como sendo do caso 0;

3.4 Base de Dados da ferramenta DB-LiOS

Em função da *Web* ser simplesmente um enorme repositório de páginas com os seus *links* embutidos descritos em *tags* HTML, faz com que a busca de páginas com as características de reuso de componentes de *links* requeira um processamento não trivial e que necessite do suporte de uma Base de Dados. Assim, a ferramenta desenvolvida neste trabalho orientou-se pela criação de uma Base de Dados para armazenar e de processos capazes de manipular as informações referentes aos *links* de um *site*.

Para assegurar todos os requisitos descritos, foi elaborada uma modelagem relacional [Silberschatz et al 1997] da Base de Dados de DB-LiOS, como pode ser observada na Figura 4.

As sumarizações das classificações 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 que são:

- Base Local ou Base de dados históricos → usada para a filtragem dos *links* das páginas dos *sites* e posterior classificação nos casos de reuso destes *links*. A base local que acompanha a instalação da ferramenta é definida no SGBD Paradox. Para se comunicar com a base em Paradox utiliza-se o BDE (*Borland Database Engine* [Borland 1999]).
- Base Global ou Base de dados de sumarizações → recebe as sumarizações das informações que foram processadas pela Base Local. É utilizada para pesquisas dos *sites* processados pelos usuários cadastrados. Esta Base Global usa PostgreSQL sobre a plataforma Linux, e é compartilhada por todos os usuários da ferramenta DB-LiOS. Para se comunicar com a base PostgreSQL utiliza-se o ODBC (*Open Database Connecting* da Microsoft [Whiting et al 1996].)

Embora o SGBD da Base Local 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 SGBDs. 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 [Seraphim & Gaspar 1998], Interbase [Interbase 1999] e PostgreSQL [PostgreSQL 1999].

Para a modelagem nos vários bancos de dados foi utilizada a ferramenta Q-Script SQL [Seraphim et al 2000]. Esta ferramenta é um W-ISQL que é um ambiente *windows* de interação de comandos SQL com qualquer bando de dados.

Figura 4 - Base de Dados Relacional implementada em DB-LiOS.

A seguir a descrição de cada tabela (relação) da Base da Dados DB-Lios

- **userlios** → Tabela que armazena os usuários do DB-LiOS. Sua chave é repassada para as tabelas *site* e *version* para identificar qual usuário cadastrou informação sobre o *site* ou a versão do *site*. Na Base de Dados Local o primeiro registro é o usuário proprietário do DB-LiOS, os demais registros são usuários que ele importou das Bases de Dados. Na Base de Dados Global estão cadastrados todos os usuários com seus respectivos identificadores únicos.
- **site** → Tabela que armazena informações sobre o *site*. Sua chave é repassada para as tabelas *version*, *pagesite*, *link* e *caselink* para identificar em qual *site* pertencem às versões, as páginas, os *links* e suas classificações. Na Base de Dados Local estão os *sites* que foram cadastrados pelo usuário. Na Base Global estão todos os *sites* que foram transferidos pelos usuários dos DB-LiOS.

- **version** → Tabela que armazena a versão do *site*, ou seja, responsável pelo controle de versões do *site*. A tabela também contém as sumarizações da classificação de *links* e páginas. Sua chave é repassada para as tabelas *pagesite*, *link* e *caselink* para identificar em qual versão pertencem às páginas, os *links* e suas classificações. Na Base de Dados Local estão as versões que foram cadastrados pelo usuário. Na Base Global estão todas as versões que foram transferidos pelos usuários dos DB-LiOS.
- **pagesite** → Tabela que armazena as páginas da versão do *site*. Sua chave é repassada para as tabelas *link* e *caselink* para identificar em qual página pertencem os *links* e suas classificações. Na Base de Dados Local estão todas as páginas das versões do *site* que foram cadastrados pelo usuário. Na Base Global essa tabela está vazia, pois esses dados não são transferidos para a Base Global para evitar excesso de tráfego na rede em um processo de transferência.
- **codehttp** → Tabela que armazena todos os códigos HTTP [Fielding 1999]. Na Base Global essa tabela está vazia.
- **link** → Tabela que armazena os *links* das páginas da versão do *site*. Sua chave é repassada para a tabela *caselink* para identificar em quais classificações pertencem os *links*. Na Base de Dados Local estão todos os *links* das páginas das versões do *site* que foram cadastrados pelo usuário. Na Base Global essa tabela está vazia, pois esses dados não são transferidos para a Base Global para evitar excesso de tráfego na rede em um processo de transferência.
- **caselink** → Tabela que armazena as classificações dos *links* das páginas da versão do *site*. Na Base de Dados Local estão todas as classificações dos *links* das páginas das versões do *site* que foram cadastrados pelo usuário. Na Base Global essa tabela está vazia, pois esses dados não são transferidos para a Base Global para evitar excesso de tráfego na rede em um processo de transferência.
- **url** → Tabela que armazena todas as URL. Um dos motivos da criação dessa tabela foi que nas estruturas de índices de SGBD, os campos indexadores não podem ser superiores a 256 caracteres. Por isso a sua chave, que é um inteiro, é repassada para a tabela *pagesite* e *link* onde esse campo é indexado. Essa indexação é utilizada nos algoritmos de classificação. Na Base Global essa tabela está vazia, pois esses dados não são transferidos para a Base Global para evitar excesso de tráfego na rede em um processo de transferência.

- **anchor** → Tabela que armazena todas as ancoras dos *links*. Um dos motivos da criação dessa tabela foi que nas estruturas de índices de SGBD, os campos indexadores não podem ser superiores a 256 caracteres. Por isso a sua chave, que é um inteiro, é repassada para a tabela *link* onde esse campo é indexado. Essa indexação é utilizada nos algoritmos de classificação. Na Base Global essa tabela está vazia, pois esses dados não são transferidos para a Base Global para evitar excesso de tráfego na rede em um processo de transferência.

3.4.1 Comunicação entre Base de Dados Local e Global

A Figura 5 mostra o sistema de comunicação entre as Base de Dados Local e Global da ferramenta DB-LiOS.

Figura 5 - Sistema de comunicação da ferramenta DB-LiOS.

A comunicação do aplicativo e a Base Global podem ser feitas:

- Por um micro situado dentro da Intranet onde está localizada a Base de Dados;
- Por um micro situado dentro de uma outra Intranet que utiliza Internet;
- Por um micro residencial que utiliza modem para acessar a Internet.

3.4.2 Transferência de dados para Base Global

A Base de Dados Global é composta por dados referentes as transferências das tabelas *site* e *version* da Base de Dados Local. Mas para que uma versão seja transferida para a Base de Dados Global não pode haver páginas não lidas e nem páginas que expiraram o tempo de espera do servidor HTTP.

3.5 Interface da ferramenta DB-LiOS

Como mencionado anteriormente a ferramenta DB-LiOS trabalha sobre 2 bases de dados: a Local e a Global. Sendo assim, a interface depende sobre qual Base de Dados está conectada.

3.5.1 Interface sobre Base Local

Na primeira vez em que a ferramenta é executada deve-se verificar a existência do usuário. Caso o usuário não exista, a ferramenta mostra a janela de usuários da ferramenta DB-LiOS (Figura 6) para que o usuário seja cadastrado.

Figura 6 - Janela de usuário da ferramenta DB-LiOS

Para a visualização de dados contidos na Base Local, os mantenedores dos *sites* (*webmasters*) possuem 7 abas de visualização implementadas em DB-LiOS, conforme descritas a seguir:

Dados sobre Sites → Na Figura 7 podem ser observadas as 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 [ISO 639-2 1998]. Para atribuição dos valores referentes ao domínio do *site* foi adotada a nomenclatura utilizada pelo *World Wide Web Corporation Consortium*.

Figura 7 - Dados sobre Sites.

Dados sobre a(s) Versão(ões) do Site → Na Figura 8 pode ser observada a informação sobre cada versão do site como 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. Pode ser observada também os seguintes sumarizadores: 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.

Site	Version	Statistics	Properties	Pages	Links	Classification
Information						
SiteName	http://www.mit.edu/					
Transfer	30/12/1999					
Model	not specified					
Link Good	52740	Total Bytes	539	1000000	Total Page 1	673
Link Bad	268	Total Links	3735	171624	Total Page 2	5
Total Links	53008	Total Page 1	3695	168024	Total Page 3	1
Page no Read	0	Total Page 2	32229	60003	Total Page 4	1
Page Unread	0	Total Page 3	3695	62744	Total Page 5	0
Page Unknown	9	Total Page 4	3735	62744	Total Page 6	1
Page Error	268	Total Page 5	3735	62744	Total Page 7	0
Page Read	268	Total Page 6	3735	62744	Total Page 8	0
Total Pages	268	Total Page 7	3735	62744	Total Page 9	0
Total Bytes	17453133	Total Page 8	3735	62744	Total Page 10	0
		Total Page 9	3735	62744	Total Page 11	0
		Total Page 10	3735	62744	Total Page 12	0

Figura 8 - Dados sobre a(s) Versão(ões) do Site.

Dados Estatísticos sobre a Versão do Site → Na Figura 9 podem ser observadas as informações estatísticas sobre cada versão do site, tais como: 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.

Site	Version	Statistics	Bytes	Links	Dead Pages
1.0.0.0	1.0.0.0	1.0.0.0	1.0.0.0	1.0.0.0	1.0.0.0
1.0.0.1	1.0.0.1	1.0.0.1	1.0.0.1	1.0.0.1	1.0.0.1
1.0.0.2	1.0.0.2	1.0.0.2	1.0.0.2	1.0.0.2	1.0.0.2
1.0.0.3	1.0.0.3	1.0.0.3	1.0.0.3	1.0.0.3	1.0.0.3
1.0.0.4	1.0.0.4	1.0.0.4	1.0.0.4	1.0.0.4	1.0.0.4
1.0.0.5	1.0.0.5	1.0.0.5	1.0.0.5	1.0.0.5	1.0.0.5
1.0.0.6	1.0.0.6	1.0.0.6	1.0.0.6	1.0.0.6	1.0.0.6
1.0.0.7	1.0.0.7	1.0.0.7	1.0.0.7	1.0.0.7	1.0.0.7

Figura 9 - Dados Estatísticos sobre a Versão do Site.

Gráfico da classificação dos links → Na Figura 10 pode ser observada uma visualização gráfica da proporção de cada caso de reuso sobre o total de *links*.

Figura 10 - Gráfico da classificação dos links.

Dados das Páginas → Na Figura 11 pode ser observada as páginas filtradas pelo *crawler* da versão do *site*. Permite visualizar as páginas: que não foram lidas pelo crawler; que expiraram o tempo de espera de resposta do Servidor HTTP; que o Servidor de HTTP retornou código http desconhecido; e as páginas que foram lidas com sucesso.

URL	Status	Tempo (ms)
http://www.mit.edu/people/jcb/jokes/	39 ok	31
http://www.mit.edu/	16 ok	24
http://www.mit.edu/forsythe-design-credit.html	0 not found	0
http://www.mit.edu/spotlight.html	0 not found	0
http://www.mit.edu/home-pages.html	12 ok	90
http://www.mit.edu/doc/how-to-hp.html	16 ok	91
http://www.mit.edu/Home-byUser.html	8 ok	24
http://www.mit.edu/alpha/home-index.html	16 ok	88
http://www.mit.edu/Home-byName.html	8 ok	24

Figura 11 - Dados das Páginas.

Dados dos Links → Na Figura 12 pode ser observado os *links* filtrados das páginas da versão do *site*. Permite filtrar as páginas: pela página-origem; ou pela ancora; ou pela página-destino.

Source URL	Anchor	Destination URL
http://www.mit.edu/people/jcb/jokes/	AltaVista	http://www.altavista.digital.com/
http://www.mit.edu/people/rei/home-links.html	AltaVista	http://www.altavista.digital.com
http://www.mit.edu/people/yonah/home.html	AltaVista	http://altavista.digital.com/
http://www.mit.edu/people/annmarie/home.html	AltaVista	http://altavista.digital.com/
http://www.mit.edu/people/jcb/	AltaVista	http://www.altavista.digital.com/
http://www.mit.edu/people/jcb/	AltaVista	http://www.altavista.digital.com/
http://www.mit.edu/people/mwhitson/mwhitson-hp	AltaVista	http://www.altavista.digital.com/
http://www.mit.edu/activities/vensamit/links.htm	AltaVista	http://altavista.digital.com/
http://www.mit.edu/people/jcb/other-instrument-jok	AltaVista	http://www.altavista.digital.com/

Figura 12 - Dados dos Links.

Classificação dos Links → Na Figura 13 pode ser observada as classes (Link0 a Link7) de casos de reuso de *links*, avaliadas naquele *site*. Permite filtrar pelos casos: link0, ou link1, ou link2 ou link3, ou link4, ou link5, ou link6, ou link7 ou todos os casos.

URL	Instrument	Page Title	Page Count
http://www.mit.edu/people/jcb/jokes/	Violin Jokes	http://www.mit.edu/people/jcb/jokes/#violin	4
http://www.mit.edu/people/jcb/jokes/	viola jokes	http://www.mit.edu/people/jcb/jokes/viola.html	7
http://www.mit.edu/people/jcb/jokes/	'Cello Jokes	http://www.mit.edu/people/jcb/jokes/#cello	3
http://www.mit.edu/people/jcb/jokes/	'Cello Jokes	http://www.mit.edu/people/jcb/jokes/#cello	4
http://www.mit.edu/people/jcb/jokes/	Bass Jokes	http://www.mit.edu/people/jcb/jokes/#bass	3
http://www.mit.edu/people/jcb/jokes/	Bass Jokes	http://www.mit.edu/people/jcb/jokes/#bass	4
http://www.mit.edu/people/jcb/jokes/	Harp Jokes	http://www.mit.edu/people/jcb/jokes/#harp	3
http://www.mit.edu/people/jcb/jokes/	Harp Jokes	http://www.mit.edu/people/jcb/jokes/#harp	4
http://www.mit.edu/people/jcb/jokes/	Piano Jokes	http://www.mit.edu/people/jcb/jokes/#piano	3

Figura 13 - Classificação dos *Links*.

3.5.2 Interface sobre Base Global

Para a visualização de dados contidos na Base Global, são disponibilizadas apenas as 4 primeira abas de visualização: Dados sobre *Sites*; Dados sobre a(s) Versão(ões) do *Site*; Dados Estatísticos sobre a Versão do *Site*; e Gráfico da classificação dos *links*. As demais abas não são visíveis, pois estariam vazias.

3.6 Funcionalidades da Ferramenta DB-LIOS

As funcionalidades da ferramenta DB-LiOS sobre cada Base de Dados também são diferentes, pois dependem de qual base de dados está conectada.

Como mostra a Figura 14 a funcionalidades da ferramenta são executadas através de menus e botões.

Figura 14 - Menu e Barra de Ferramentas do DB-LiOS

3.6.1 Funcionalidades sobre Base Local

As funcionalidades sobre a Base Local são descritas abaixo. Os botões são classificados em botões de navegação e botões de operação sobre a base de dados. Os botões de navegação permitem a movimentação entre os registros das tabelas *site* e *version*, dependendo da aba que está ativa:

- First → Move para o primeiro registro da tabela *site* ou para o primeiro registro da tabela de versão do *site*;
- Back → Move para o registro anterior na tabela *site* ou para o registro anterior na tabela de versão do *site*;
- Next → Move para o próximo registro na tabela *site* ou para o próximo registro na tabela de versão do *site*;
- Last → Move para o último registro da tabela *site* ou para o último registro da tabela de versão do *site*;

Os botões de operação sobre a base de dados realizam as operações de cadastro sobre a tabela *site* e *versão*, dependendo da aba que está ativa:

- Add → Adiciona novo *site* ou nova versão do *site*. Para adicionas um novo *site* ou versão deve-ser preencher o formulário da Figura 15;
- Edit → Altera dados sobre o *site* ou da versão do *site*;
- Delete → Deleta uma versão de um *site* ou um *site* com todas as suas versões;
- Restart → Reinicia a filtragem de *links* das páginas do *site*, conforme foi mostrado no final da seção *Crawler*;
- Refresh → Atualiza os dados do aplicativo DB-LiOS. Essa função é necessária pois vários usuários podem estar compartilhando informações de uma mesma base de dados. É necessário atualizar os dados visualizados pelo aplicativo com a base de dados.

Figura 15 - Tela para adicionar um novo site ou uma nova versão do *Site*

A Figura 16 mostra o menu *File* da ferramenta DB-LiOS que possui as seguintes funcionalidades:

- Info User → Visualiza e permite alterar informações sobre o usuário;
- Connected Global / Connected Local → Muda a conexão com a base de dados entre Global e Local;
- Transfer Global → Transfere os dados da Base Local para a Base Global conforme foi mostrado no final da seção Transferência de dados para Base Global;
- Options → Permite configurar propriedades do *crawler*, tais como: tempo de espera de resposta do servidor http; tempo de intervalo entre as solicitações a um mesmo servidor http; e número máximo de solicitações a uma mesma URL;
- Exit → Sai da ferramenta DB-LiOS.

Figura 16 - Menu *File* da ferramenta DB-LiOS

A Figura 17 mostra o menu *Operation* da ferramenta DB-LiOS que possui as mesmas funcionalidades que a barra de ferramentas.

Figura 17 - Menu *Operation* da ferramenta DB-LiOS

3.6.2 Funcionalidades sobre Base Global

As funcionalidades sobre a base global são os botões de navegação que dependem da aba que está ativa, seja ela a aba Site ou Version (First, Back, Next e Last). Os botões de operação sobre a base de dados ficam todos desativados, menos o botão Refresh.

3.7 Ferramenta DataImport

A Figura 18 mostra a tela da ferramenta DataImport. A ferramenta permite importar a todas os dados das tabelas da Base Local de outro usuário.

Figura 18 - Ferramenta DataImport

Para importar os dados das tabelas da Base Local basta copiar o diretório Data que está dentro do diretório onde foi instalado a ferramenta do outro usuário para a máquina que vai receber os dados. No botão *Open* localize esse diretório e clique no botão OK para iniciar a importação dos dados. Mas somente dados que foram transferidos para a Base Global podem ser transferidos.

A utilidade dessa ferramenta é unir numa mesma Base de Dados Local extrações/classificações de *sites* que foram realizados por vários usuários.

3.8 Considerações Finais

Para a implementação da primeira versão da ferramenta foram dedicados cinco meses de desenvolvimento. Após o término dessa primeira versão, alunos da graduação na disciplina Hipermídia (SCE-225) do curso de curso de Bacharelado em Ciência da Computação do ICMC/USP, utilizaram a ferramenta para analisar a reusabilidade dos *links* em *sites* de instituições educacionais de curso superior.

A partir desse experimento, novas necessidades foram especificadas conforme descritas na seção Histórico deste capítulo. Dessa forma, quatro outros meses foram dedicados para que a ferramenta se encontrasse no estado autal de desenvolvimento. Hoje a ferramenta está totalmente implementada e disponível para *download* no endereço <http://dblios.icmc.sc.usp.br/> para que pesquisadores e *webmasters* a utilizem para analisar a reusabilidade das páginas. No próximo capítulo são descritos experimentos realizados com a ferramenta DB-LiOS.

4. 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 2. Esses experimentos visavam validar a proposta de que a ferramenta DB-LiOS pode auxiliar a tarefa de coletar as métricas propostas em [Fortes et al 1997]. 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.

4.1 Descrição dos Experimentos Realizados

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 2).

A Tabela 3 mostra resultados sobre os *links* dos *sites*. 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 19.

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 Tabela 4 mostra resultados sobre as sumariações de classificações dos *links* sobre cada site.

Tabela 2 - Resultados sobre os sites obtidos pelo DB-LiOS.

URL	Início Extração	Bytes Lidos	Páginas		
			Mortas	Boas	Total
http://www.anhembi.br/	29/02/2000 00:09	1.777.803	6	129	135
http://www.baraodemaua.br/	29/02/2000 20:39	387.562	2	180	182
http://www.bmcc.org/	01/03/2000 02:02	3.105.494	137	411	548
http://www.cam.ac.uk/	15/03/2000 17:35	33.448.886	470	5.493	5.963
http://www.dcc.unicamp.br/	29/02/2000 12:13	1.032.966	2	78	80
http://www.eca.usp.br/	02/03/2000 14:41	21.272	0	29	29
http://www.eesc.sc.usp.br/	02/03/2000 12:21	963.015	5	273	278
http://www.esalq.usp.br/	15/03/2000 16:52	773.082	2	71	73
http://www.fcf.usp.br/	15/03/2000 17:03	611.204	6	80	86
http://www.fcfrp.usp.br/	02/03/2000 15:07	426.537	1	51	52
http://www.fe.usp.br/	15/03/2000 16:37	3.314.625	10	300	310
http://www.fea.usp.br/	02/03/2000 14:42	7.022.102	16	537	553
http://www.fee.unicamp.br/	29/02/2000 00:21	2.499.609	25	339	364
http://www.fob.usp.br/	15/03/2000 17:23	3.132	0	24	24
http://www.harvard.edu/	02/03/2000 01:17	506.486	0	120	120
http://www.hokudai.ac.jp/	02/03/2000 00:28	4.660.685	18	1.272	1.290
http://www.ib.usp.br/	02/03/2000 16:42	2.794.077	27	580	607
http://www.icb.usp.br/	02/03/2000 17:55	245.324	0	37	37
http://www.icmc.sc.usp.br/	02/03/2000 13:39	1.367.589	5	143	148
http://www.if.sc.usp.br/cgi-bin/w3- mysql/Gera_Home-Page.htm	02/03/2000 13:50	38.275	0	288	288
http://www.if.usp.br/	02/03/2000 14:06	1.352.913	129	581	710
http://www.igc.usp.br/parte2.html	15/03/2000 16:24	946.702	2	216	218
http://www.ime.usp.br/	02/03/2000 14:24	4.924.199	13	647	660
http://www.io.usp.br/	02/03/2000 14:37	179.289	1	80	81
http://www.iqsc.sc.usp.br/	02/03/2000 13:59	1.501.237	0	230	230
http://www.mit.edu/	29/02/2000 01:16	17.493.183	218	3.675	3.893
http://www.poli.usp.br/	15/03/2000 16:36	24.591	0	41	41
http://www.puc-rio.br/	01/03/2000 01:11	3.358.580	6	1.078	1.084
http://www.udmercy.edu/	01/03/2000 00:22	2.254.060	93	810	903
http://www.uerj.br/	29/02/2000 11:58	274.345	1	30	31
http://www.ufba.br/	30/03/2000 23:02	13.056.960	32	863	895
http://www.ufes.br/	29/02/2000 18:52	1.748.372	10	311	321
http://www.ufpr.br	29/02/2000 12:01	178.237	0	73	73
http://www.ufrgs.br/	26/02/2000 01:08	3.714.289	43	697	740
http://www.ufrj.br/	29/02/2000 23:54	287.175	1	106	107
http://www.ufrn.br/	27/03/2000 15:06	1.132.095	8	83	91
http://www.unesp.br/home.htm	26/02/2000 00:42	2.682.630	9	144	153
http://www.unicamp.br/	29/02/2000 09:02	82.153	4	6	10
http://www.unifenas.br/	29/02/2000 10:06	757.248	9	230	239
http://www.unimep.br/	26/02/2000 00:31	329.102	0	35	35
http://www.usp.br/	29/02/2000 23:43	492.406	1	197	198
TOTAL		121.769.491	1.312	20.568	21.880

Tabela 3 - Resultados sobre os *links* dos sites obtidos pelo DB-LIOS.

URL	Links					
	Não Lidos	Timeout	Desco- nhecido	Mortos	Bons	Total
http://www.anhembi.br/	0	0	32	0	6.169	6.137
http://www.baraodemaua.br/	0	0	0	0	515	515
http://www.bmcc.org/	0	0	0	1	294	293
http://www.cam.ac.uk/	0	0	0	2	1.895	1.893
http://www.dcc.unicamp.br/	0	0	0	0	155	155
http://www.eca.usp.br/	0	0	1	35	9.633	9.597
http://www.eesc.sc.usp.br/	0	0	0	6	6.324	6.318
http://www.esalq.usp.br/	0	0	0	39	5.745	5.706
http://www.fcf.usp.br/	0	0	0	0	935	935
http://www.fcfrp.usp.br/	0	0	0	2	257	255
http://www.fe.usp.br/	0	0	0	9	2.542	2.533
http://www.fea.usp.br/	0	0	0	0	501	501
http://www.fee.unicamp.br/	0	0	1	15	1.496	1.480
http://www.fob.usp.br/	0	0	0	1	404	403
http://www.harvard.edu/	0	0	0	39	6.861	6.822
http://www.hokudai.ac.jp/	0	0	0	37	1.774	1.737
http://www.ib.usp.br/	0	0	0	58	1.464	1.406
http://www.icb.usp.br/	0	0	3	180	7.298	7.115
http://www.icmc.sc.usp.br/	0	0	0	177	5.874	5.697
http://www.if.sc.usp.br/cgi-bin/w3- mysql/Gera_Home-Page.htm	0	0	0	41	2.795	2.754
http://www.if.usp.br/	0	0	1	7	867	859
http://www.igc.usp.br/parte2.html	0	0	0	2	1.348	1.346
http://www.ime.usp.br/	0	0	4	0	66	62
http://www.io.usp.br/	0	0	0	175	3.245	3.070
http://www.iqsc.sc.usp.br/	0	0	0	31	728	697
http://www.mit.edu/	0	0	0	0	108	108
http://www.poli.usp.br/	0	0	0	0	157	157
http://www.puc-rio.br/	0	0	0	4	1.154	1.150
http://www.udmercy.edu/	0	0	0	7	425	418
http://www.uerj.br/	0	0	0	5	1.612	1.607
http://www.ufba.br/	0	0	0	0	47	47
http://www.ufes.br/	0	0	0	46	5.017	4.971
http://www.ufpr.br	0	0	9	265	53.279	53.005
http://www.ufrgs.br/	0	0	26	1.304	86.609	85.279
http://www.ufrj.br/	0	0	0	5	2.164	2.159
http://www.ufm.br/	0	0	0	0	422	422
http://www.unesp.br/home.htm	0	0	0	6	7.237	7.231
http://www.unicamp.br/	0	0	0	116	6.263	6.147
http://www.unifenas.br/	0	0	0	16	3.615	3.599
http://www.unimep.br/	0	0	0	0	3.585	3.585
http://www.usp.br/	0	0	0	1	1.085	1.084
TOTAL	0	0	77	2.632	241.964	239.255

Tabela 4 - Resultados sobre as classificações dos *links* dos sites obtidos pelo DB-LiOS

URL	Classificação de Links							
	link0	link1	link2	link3	link4	link5	link6	link7
http://www.anhembi.br/	0	8	0	7	56	6	0	0
http://www.baraodemaua.br/	1	189	662	1.200	2.891	117	41	33
http://www.bmcc.org/	2	0	1	102	91	64	0	0
http://www.cam.ac.uk/	4	53	30	350	511	102	4	6
http://www.dcc.unicamp.br/	13	250	36	765	1.427	54	8	88
http://www.eca.usp.br/	0	273	287	484	1.375	348	6	8
http://www.eesc.sc.usp.br/	35	7.456	5.610	38.934	53.497	6.659	2.035	22.727
http://www.esalq.usp.br/	1	158	75	927	1.194	85	22	56
http://www.fcf.usp.br/	1	142	163	551	1.833	158	19	145
http://www.fcfrp.usp.br/	28	713	267	3.379	6.093	609	188	129
http://www.fe.usp.br/	0	189	47	950	1.362	1.169	2.498	2.305
http://www.fea.usp.br/	0	27	0	47	102	48	2	2
http://www.fee.unicamp.br/	0	92	68	70	427	57	0	27
http://www.fob.usp.br/	0	59	24	59	150	45	31	18
http://www.harvard.edu/	7	7	0	2	35	2	0	2
http://www.hokudai.ac.jp/	1	85	53	512	531	187	451	169
http://www.ib.usp.br/	11	1.025	469	3.352	5.395	782	437	579
http://www.icb.usp.br/	2	65	13	59	258	93	28	57
http://www.icmc.sc.usp.br/	2	23	4	48	182	50	252	20
http://www.ifsc.usp.br/cgi-bin/w3-mysql/Gera_Home-Page.htm	3	680	88	1.480	2.410	222	155	60
http://www.if.usp.br/	0	99	0	72	388	32	0	0
http://www.igc.usp.br/parte2.html	2	282	249	3.242	4.279	780	83	536
http://www.ime.usp.br/	0	190	22	11	235	8	154	2
http://www.io.usp.br/	0	36	14	2.263	2.118	386	8	22
http://www.iqsc.sc.usp.br/	5	137	35	319	784	64	42	18
http://www.mit.edu/	9	726	338	3.693	5.701	980	2.199	691
http://www.poli.usp.br/	1	787	186	1.936	3.195	322	40	22
http://www.puc-rio.br/	7	759	210	2.335	4.395	716	205	692
http://www.udmercy.edu/	59	3.796	3.656	32.229	43.961	4.504	1.116	3.286
http://www.uerj.br/	0	94	7	810	828	220	30	9
http://www.ufba.br/	6	451	419	3.732	5.067	706	104	265
http://www.ufes.br/	0	58	3	77	194	29	6	62
http://www.ufpr.br	1	21	3	12	87	11	6	0
http://www.ufrgs.br/	11	125	136	249	1.598	55	22	124
http://www.ufjf.br/	1	527	324	2.286	2.865	628	69	38
http://www.ufrn.br/	11	860	106	1.621	4.721	502	156	349
http://www.unesp.br/home.htm	0	81	46	387	666	123	185	178
http://www.unicamp.br/	0	30	58	614	731	55	40	28
http://www.unifenas.br/	0	85	19	882	1.345	97	8	13
http://www.unimep.br/	7	594	209	2.090	3.755	821	107	217
http://www.usp.br/	6	1.143	655	2.722	4.928	970	330	401
TOTAL	237	22.375	14.592	114.860	171.661	22.866	11.087	33.384

A Figura 19 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 19 - Percentuais médios dos casos de reuso de *links* dos sites obtidos pelo DB-LiOS

Observou-se também uma maior ocorrência na classificação dos *links* nos casos *link3* e *link4*. Os *links3* são os que possuem mesma âncora em páginas-origem diferentes e apontam para uma mesma página-destino. Os *links4* 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 *link4* (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 *link3* (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 (*link1*, *link2*, *link5* e *link6*) 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 .

A Tabela 5 mostra a totalização dos casos de páginas obtidos pelo DB-LiOS. Podemos agrupar os valores da Tabela 5 nos 3 grupos de páginas: as que não oferecem navegação; as que oferecem navegação linear; e as que oferecem navegação com multiplicidade de escolhas.

Páginas que não oferecem navegação são páginas que não possuem *links*. Essas páginas, também conhecidas como página folha, foram classificadas como Page 0. Totalizaram 3.350 páginas, pois das 4.739 classificadas como Page 0, 1.312 são páginas mortas e 77 são páginas que retornaram código de http desconhecido. Isso demonstra que 15,31% das páginas destes sites não utilizam da principal característica encontrada no hiperdocumento que é a navegação.

Páginas que oferecem navegação linear são páginas que possuem apenas 1 *link*. Fazem parte desse grupo as classificações de páginas Page 16, Page 32, Page 48, Page 64, Page 80, Page 96, Page 112 e Page 128. Totalizando 2.207 páginas que apresentam essa característica. Isso demonstra que 10,73% das páginas desses *sites* o autor limitou a navegação do usuário, forçando ao usuário navegar linearmente.

Páginas que oferecem navegação com multiplicidade de escolhas são páginas, ou seja, páginas que possuem 2 ou mais *links*. Todas as demais classificações são páginas apresentam multiplicidade de escolha. Totalizando 15.011 páginas que apresentam essa característica. Isso demonstra que 73,96% das páginas dos *sites* o autor oferece multiplicidade de opções para a navegação do usuário.

A classificação de páginas Page24 (*link3* + *link4*) é o que mais aparece nos sites cerca de 18,69%, isso se deve ao fato que os *links* classificados como *link3* e *link4*, são 73,27% dos casos de *links*.

Tabela 5 - Totalização dos casos de páginas obtidos pelo DB-LiOS

Caso da Página	Total
Page0	4739
Page24	4090
Page120	1643
Page28	1434
page16	1406
page88	1294
page56	630
page124	570
page72	551
page92	443
page60	413
page125	390
page64	319
page8	302
page104	268
page121	265
page29	258
page128	237
page25	167
page93	155
page32	130
page40	127
page18	120
page96	115
page61	101
page20 e page 89	99
page127	93
page26	80
page76	75
page126	72
page122	68
page31	67
page1	66
page30	65
page36	63
page94	61
page57	59
page90	58
page108	52

Caso da Página	Total
page123	40
page12	39
page95	35
page62 e page73	30
page74	29
page58	28
page4	27
page42	25
page44	24
page105	23
page10	21
page9 e page63	20
page2	19
page52 e page66	18
page21	15
page106	14
page109	13
page13 e page77	12
page41, page78 e page91	11
page45 e page107	9
page3 e page110	7
page5	5
page11, page14, page23, page37, page 59, page79 e page111	4
page22 e page47	3
page15, page27, page43, page67, page75, page82 e page86	2
page6, page7, page19, page39, page54 e page70	1
page17, page33, page34, page35, page38, page46, page48, page49, page50, page51, page53, page55, page65, page68, page69, page71, page80, page81, page83, page84, page85, page87, page97, page98, page99, page100, page101, page102, page103, page112, page113, page114, page115, page116, page117, page118 e page119	0

O Page127 é um caso muito especial pois nele aparece todas as possibilidades de páginas - menos Page0 porque se ele aparece em uma página não pode aparecer nenhum outro. Assim, as páginas classificadas como Page127, são as mais ricas em diversidade de reusabilidade dos componentes do *link*, pois apresentam o máximo de casos de *links*. Estas páginas apresentam: $\text{Link1 (64) + Link2 (32) + Link3 (16) + Link4 (8) + Link5 (4) + Link6 (2) + Link7 (1) = 127}$. Embora seja difícil de ele ocorrer, nesta amostragem aparece 93 vezes, ou 0,425%.

Nos sites avaliados, 37 classificações de páginas não foram encontradas, que são os casos: page17, page33, page34, page35, page38, page46, page48, page49, page50, page51, page53, page55, page65, page68, page69, page71, page80, page81, page83, page84, page85, page87, page97, page98, page99, page100, page101, page102, page103, page112, page113, page114, page115, page116, page117, page118 e page119.

Como esses casos não foram encontrados em uma amostra de 20.568 páginas válidas, um estudo mais detalhado deve ser feito para verificar se é possível ou não uma página ser classificada em um desses casos.

4.2 Considerações Finais

Esses experimentos são uma primeira amostra da utilização da ferramenta DB-LiOS. Os 41 *sites* de instituições de ensino superior analisados mostra um esforço em analisar reusabilidade em sites com esta característica. No futuro novas extrações/classificações poderão ser executadas para observar a evolução desses sites.

No próximo capítulo apresentamos algumas ferramentas que analisam algumas características na *Web* bem como uma análise entre elas.

5. Ferramentas de avaliação de *Websites*

É notável a evolução tecnológica que tem propiciado uma rápida popularização no uso da *Web*. Em contrapartida, observa-se que, muitas vezes, essa popularidade leva a hiperdocumentos mal "escritos" que não podem ser completamente testados nem adequadamente mantidos [Brown 1990].

Na *Web*, devido à liberdade de autoria oferecida pela linguagem HTML por meio de simples marcações (*tags*) identificadores, hiperdocumentos "mal estruturados" se tornam comuns. No entanto, esses mesmos *tags* trazem também a solução do problema. No caso da definição de *links*, os *tags* `<a>` e `</a>`, 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 [Berners-Lee et al. 1994].

Com o objetivo de reduzir a complexidade da manutenção de hiperdocumentos da *Web* muitas pesquisas têm sido realizadas. Entre elas, ferramentas que auxiliam a avaliação e medição de atributos de hiperdocumentos da *Web* tem sido desenvolvidas. Este capítulo apresenta algumas ferramentas comerciais disponíveis que se propõem a avaliar *Websites*, especificamente com relação aos seus *links*.

5.1 Ferramentas de Avaliação

Nesta seção descrevemos três ferramentas de avaliação de *links* na *Web*: Linkbot Enterprise Server 2.0 [Watchfire 2000], WebAnalyser 1.1 [Incontext 2000], LinkProver 2.1b [TaFWeb 2000].

5.1.1 Ferramenta Linkbot Enterprise Server 2.0

A ferramenta Linkbot Enterprise Server 2.0 [Watchfire 2000] como mostra a Figura 20, é um software comercial que visita todas as páginas que compõem um *site* e analisa um conjunto de características.

Figura 20 - Tela do Linkbot Enterprise Server 2.0

Para visitar todas as páginas, as mesmas devem estar endereçadas por URL, ou seja dentro de um Servidor HTTP. A ferramenta não verifica as páginas dentro de sistema de arquivos (*File System*).

A ferramenta Linkbot exige: IBM PC/Windows 95 ou superior, conexão Internet, 32 Mb Memória RAM e a base de dados consome em média 2.5Kb por página, ou seja, 2.5Mb a cada mil páginas. As informações extraídas das páginas do *site* são armazenadas em uma base de dados gerenciada pelo banco de dados SYBASE ANYWHERE. A ferramenta possui um desempenho muito bom, o seu tempo de execução é extremamente rápido, pois confere ao mesmo tempo até 30 URLs (30 threads). Permite também agendar um horário para a extração dos *sites*.

Durante a visita das páginas, a ferramenta Linkbot Enterprise Server 2.0 analisa as seguintes características:

- existência da página que o *link* está referenciando;
- existência das imagens que a página contém;
- existência de título nas páginas;
- arquivos órfãos, ou seja, arquivos que não são utilizados pelo Servidor de *WEB*;

- sintaxe válida de e-mails;
- *links* em Java Script;
- *tags* de formulários (FORM);
- se os atributos *width* e *height* (largura e altura) das imagens foram definidos nas páginas, pois o *browser* reserva o espaço do tamanho da imagens, sem ter que empurrar o texto para baixo;
- páginas grandes que podem ser lentas para carregar.

Os resultados da análise são armazenados em páginas HTML.

5.1.2 Ferramenta WebAnalyser 1.1

A ferramenta WebAnalyser 1.1 [Incontext 2000] como mostra a Figura 21, também é um software comercial que visita todas as páginas que compõem um *site* e analisa um conjunto de características. Também utiliza os endereços de URL para visitar todas as páginas que compõem o *site*. A ferramenta não verifica as páginas dentro de sistema de arquivos (*File System*).

A ferramenta Linkbot exige: IBM PC/Windows 95 ou superior, conexão Internet, 16 Mb Memória RAM. As informações extraídas das páginas do *site* são armazenadas em uma base de dados gerenciada pela própria ferramenta. A ferramenta permite também criar uma cópia local do *site* analisado, criando assim uma cópia espelho do *site*. Mas a principal vantagem dessa ferramenta é a visualização gráfica. Ela oferece cinco tipos de visões interativas:

1. *Wavefront* → as páginas são dispostas em círculos concêntricos, como mostra a caixa de rolagem superior direita da Figura 21. A medida que um nível é percorrido no *site* um novo círculo é adicionado;
2. *File View* → é uma grade, como mostra a caixa de rolagem inferior da Figura 21, que exhibe informações: nome da URL, tipo do arquivo, tamanho do arquivo, nível no *site*, data, título do documento.
3. *URL Tree* → é uma árvore onde são dispostas todas as URLs do site;
4. *Site Map* → é uma árvore onde são dispostos todos os elementos que compõem o *site*;
5. *Link views* → um ambiente gráfico, como a mostra caixa de rolagem superior esquerda da Figura 21, que mostra a partir de uma página *n* selecionada: as páginas que referenciam a página *n* → página *n* → as páginas que são referenciadas pela página *n*.

Figura 21 - Tela da ferramenta WebAnalyser 1.1

Durante a visita das páginas, a ferramenta WebAnalyser 1.1 analisa as seguintes características:

- existência da página que o *link* está referenciando;
- existência das imagens que a página contém;
- existência de arquivos duplicados;

Os resultados da análise são armazenados em páginas HTML.

5.1.3 Ferramenta LinkProver 2.1b

A ferramenta LinkProver 2.1b [TaFWeb 2000] apresenta-se como mostra a Figura 22. Trata-se de um software "free" (disponível livremente para *download*) que verifica um conjunto de características de uma única página. Ela usa tanto os endereços de URL das páginas, como também o sistema de arquivos (*File System*) para acessar as páginas.

Figura 22 - Tela da ferramenta LinkProver 2.1b

A ferramenta LinkProver 2.1b exige: IBM PC/Windows 95 ou superior, conexão Internet, 16 Mb Memória RAM.

Durante a visita da página, a ferramenta LinkProver 2.1b analisa as seguintes características:

- existência da página que o *link* está referenciando;
- existência das imagens que a página contém;

Os resultados da análise são armazenados em páginas HTML.

5.2 Análise das Ferramentas

A Tabela 6 mostra uma análise das funcionalidades estudadas nas ferramentas Linkbot Enterprise Server 2.0 [Watchfire 2000], WebAnalyser 1.1 [Incontext 2000], LinkProver 2.1b [TaFWeb 2000] e DB-LiOS 1.0 [Seraphim & Fortes 2000] que foi desenvolvida neste trabalho.

Tabela 6 - Análise das funcionalidades entre as ferramentas de avaliação

Funcionalidade	Linkbot	WebAnalyser	LinkProver	DB-LiOS
Verifica existência da página que o <i>link</i> está referenciando	Sim	Sim	Sim	Sim
Verifica existência das imagens que a página contém	Sim	Sim	Sim	Não
Verifica existência de título nas páginas	Sim	Não	Não	Não
Verifica existência de arquivos duplicados	Não	Sim	Não	Não
Verifica sintaxe de e-mails	Sim	Não	Não	Não
Verifica arquivos órfãos (arquivos não utilizados pelo Servidor de <i>WEB</i>)	Sim	Não	Sim	Não
Verifica se os atributos <i>width</i> e <i>height</i> das imagens estão definidos	Sim	Não	Não	Não
Verifica a reusabilidade de <i>links</i> e páginas	Não	Não	Não	Sim
Cria cópias espelho do site	Não	Sim	Não	Não
Apresenta vários ambientes de visualização dos elementos do <i>site</i>	Não	Sim	Não	Não
Realiza leitura das páginas a partir endereços de URL	Sim	Sim	Sim	Sim
Realiza leitura das páginas a partir de sistema de arquivos	Não	Não	Sim	Não

A Tabela 7 mostra uma análise das características de tempo de execução das ferramentas de avaliação estudadas. Todas as ferramentas foram executadas considerando-se uma avaliação do *site* do Laboratório de Engenharia de Software do ICMC-USP, ou seja, a URL "<http://nt-labes.icmc.sc.usp.br/>" foi fornecida como entrada.

Tabela 7 - Análise dos experimentos entre as ferramentas de avaliação

Característica	Linkbot	WebAnalyser	LinkProver	DB-LiOS
Total de páginas lidas	254	197	-	92
Total <i>links</i> mortos (internos)	20	17	-	13
Total de imagens mortas	5	15	-	-
Total de arquivos duplicados	-	6	-	-
Tamanho da base de dados	3.28Mb	356Kb	-	1.44Mb

A ferramenta DB-LiOS leu um quantidade de páginas inferiores as demais ferramentas avaliadas, pois DB-LiOS verifica somente os *links* que são formados pelos tags `<a>` `</a>`. As demais ferramentas também verificam *links* formados por Java Script.

5.3 Considerações Finais

Neste capítulo foram apresentadas as ferramentas Linkbot, WebAnalyser e LinkProver, que analisam estruturas de documentos de *sites* da *Web*.

A partir de experimentos realizados com a execução das mesmas, foi possível identificar características e funcionalidades desejáveis para o desenvolvimento de uma ferramenta que possibilite automaticamente avaliar documentos na *Web*.

Concluimos também que existem várias ferramentas no mercado que analisam características de documentos na *Web*, mas cada uma delas se concentra em determinadas características. Dessa forma, uma avaliação "de fato" deve sempre considerar uma análise complementar de um *website* por meio da obtenção de resultados de mais de uma dessas ferramentas.

Neste trabalho, a característica de reusabilidade estrutural é focalizada. O próximo capítulo apresenta as conclusões e trabalhos futuros sobre este projeto.

6.1 Conclusões Finais

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* [Seraphim & Fortes 2000]. Além disso, os resultados obtidos com essa classificação mostraram que podem auxiliar os *webmasters* a verificar se os *links* de seu *site* que ajudam o usuário na navegação [Fortes et al 1997]. 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*.

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*.

Os primeiros estudos realizados com a classificação de páginas permitiram estabelecer relações entre os casos de páginas e tipo de navegação: linear, não linear, ou não oferece navegação.

A partir dos experimentos pode-se concluir também que a maioria das páginas não oferecem navegação ao usuário. Essas páginas não se utilizam da principal característica encontrada no hipertexto que é a navegação.

6.2 Trabalhos futuros

Trabalhos futuros podem ser investigados em 3 áreas:

➤ *Casos de reusabilidade de links e de páginas*

Estudos podem ser realizados na Base de Dados Global no sentido de se buscar alguma uniformidade na classificação de *links* e páginas de *sites* com o mesmo domínio. Também pode-se investigar se através da classificação de *links* e de páginas, existem outras características a serem evidenciadas. As 37 classificações de páginas que não foram encontradas nos experimentos com a ferramenta DB-LiOS devem ser estudadas com mais detalhe, para se verificar se é possível ou não uma página obter uma dessas classificações.

➤ *Crawler*

O *crawler* da ferramenta DB-LiOS poderia ser adaptado para que, enquanto está varrendo as páginas do site, além de extrair *links*, se verifique outros requisitos como:

- existência das imagens que a página contém;
- existência de páginas externas ao *site*;
- existência de arquivos duplicados;
- existência de e-mails;
- existência de título nas páginas;
- arquivos órfãos, ou seja, arquivos que não são utilizados pelo Servidor de *WEB*;
- *links* em Java Script e XML;
- se os atributos *width* e *height* (largura e altura) das imagens foram definidos nas páginas.

➤ *Ferramenta DB-LiOS*

Com a estrutura extraída pelo *crawler* poderia-se criar um modelo gráfico de visualização dos elementos que formam o *site* proporcionando uma melhor interface dos usuários com a ferramenta. Pode-se ainda criar um sistema de consultas para as Bases de Dados Local e Global e ainda um sistema de agendamento de horários para que os *sites* sejam analisados. Poderia-se também criar uma cópia local do *site* analisado, por meio de uma cópia espelho do *site*.

7. Referências Bibliográficas

- [Begoray 1990] Begoray, J.A., "An Introduction to Hypermedia Issues, Systems and Application Areas" *Intern. Journal Man-Machine Studies*, v.33, p.121-47, 1990.
- [Berners-Lee et al. 1994] Berners-Lee, T. et al., "The World-Wide Web" *Comm. of the ACM*, v.37, n.8, p.76-82, 1994.
- [Bernstein, M 1990] Bernstein, M. "An Apprentice That Discovers Hypertext Links" *Hypertext: Concepts, Systems and Applications* Cambridge, Great Britain, Cambridge University Press, 1990. p.212-23.
- [Blue 1997] Blue, T.; *Development Database in Delphi*. First Edition, Makron Books, 1997.
- [Borland 1999] Borland Corp., "Borland Database Engine - BDE", online junho/1999, disponível na internet: http://www.borland.com/devsupport/bde/ti_list/.
- [Botafofo & Shneiderman 1992] Botafofo, 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.
- [Brown 1990] Brown, P. J. Assessing the Quality of Hypertext Documents. In: European Conference on Hypertext, Versailles, França, novembro 1990. *Proceedings*. p.1-12.
- [Conklin 1987] Conklin, J., "Hypertext: An Introduction and Survey" *IEEE Computer*, v.20, n.9, p.17-41, 1987.
- [Cho et al 1998] Cho, J.; Garcia-Molina, H.; Page L.; "Efficient crawling through URL ordering" in *proc. 7th Intern. WWW Conference*, Brisbane, Australia. April, 1998.
- [Fielding 1999] 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>.
- [Fortes & Nicoletti 1997] Fortes, R.P.M.; Nicoletti, M.C. "A Family of Link Based Metrics for the Evaluation of Web Documents" *In SIGLINK Newsletter*, V. 6, No. 3, p.21-23, 1997.

- [Fortes et al 1997] Fortes, R. P. M.; Nicoletti, M.C.; Neto, A., "A Formal Approach to Consistency and Reuse of *Links* in World Wide Web Applications" In *Formal Methods in Human-Computer Interaction*, Philippe Palanque & Fabio Paternò (eds.), Chapter 4, Springer-Verlag, p.75-92, 1997.
- [Fortes 1996] 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.
- [Furnas 1986] Furnas, G.W. "Generalized fisheye views" In: ACM CHI'86 *Human Factors in Computing Systems*, Boston, April 1986. *Proceedings*. 1986. p.16-23.
- [Garzotto et al 1995] Garzotto, F.; Mainetti, L.; Paolini, P., "Hypermedia Design, Analysis and Evaluation Issues" *Comm. of the ACM* v. 38, n. 8, p.74-96, 1995.
- [Halasz 1988] 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.
- [Halasz & Schwartz 1990] 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.
- [Hardman 1995] 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.
- [Hatzimanikatis et al 1995] 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.
- [Incontext 2000] Incontext Systems, "WeAnalyzer 2.01", online fevereiro/2000, disponível na internet: <http://www.incontext.com/WAinfo.html>.
- [Interbase 1999] Interbase, "Interbase: The OPEN source database", online junho/1999, disponível na internet: <ftp://ftp2.interbase.com/pub/products/release/Windows/id55docs.zip>
- [ISO 639-2 1998] Technical contents of ISO 639-2/1998 *Code for the representation of names of languages*: Part 2: Alpha-3 code.

- [Juran & Gryna Jr 1970] Juran, J.M.; Gryna Jr, F.M. *Quality Planning and Analysis: From Product Development Through Use*. New York USA, McGraw-Hill, 1970.
- [Kollar 1999] Kollar, C.P.; Leavitt, J.R.R.; Mauldin, M. "Robot Exclusion Standard Revisted", online julho/1999, disponível na internet: <http://www.kollar.com/robots.html>..
- [Koster 1999] Koster, M., "Guidelines for Robot Writes" online junho/1999, disponível na internet: <http://info.webcrawler.com/mak/projects/robots/guidelines.html>.
- [Koster 1995] Koster, M., "Robots in the Web: threat or treat?", *ConneXions*, V. 9, No. 4, April 1995.
- [Marshall & Shipman 1993] 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.
- [Nielsen 1990] Nielsen, J. "The Art of Navigating through Hypertext" *Comm. of the ACM*, v.33, n.3, p.296-310, March 1990.
- [Nielsen 1995] Nielsen, J., *Multimedia and Hypertext - The Internet and Beyond*. Academic Press, London, United Kingdom, 1995.
- [Pinkerton 1994] Pinkerton, B. "Finding what people want: experiences with the WebCrawler", In *Proc. of the 2nd International WWW Conference*, Chicago, USA, October 17–20, 1994.
- [PostgreSQL 1999] PostgreSQL, "PostgreSQL TODO list", on-line junho/1999, disponível na internet: <http://www.postgresql.org/docs/todo.html>
- [Salton et al 1994] 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.
- [Schwabe et al 1996] 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.
- [Seraphim & Gaspar 1998] Seraphim, E.; Gaspar, D. A. *Microsoft Access 97 - Suport SQL*. São Paulo, 1998. Technical Report N° 79 - USP.
- [Seraphim & Fortes 1999] Seraphim, E.; Fortes, R. P. M. "Programação Delphi apoiando a construção de aplicativos de bases de dados", *Série Notas Didáticas do ICMC No 42*, dezembro de 1999, 61p.

- [Seraphim et al 2000] Seraphim, E.; Gaspar, D. A.; Traina, C. Junior (1999) " Q-Script - A tool to Move Data Between two Relational Databases", VII Conference on Extending Database Technology - EDTB 2000, University of Konstanz, Alemanha, Março de 2000.
- [Seraphim & Fortes 2000] Seraphim, E.; Fortes, R. P. M. "Ferramenta DB-LiOS para Avaliação de Reuso de *links* em *WWW*" *Aceito para Apresentação no XX Congresso Nacional da Sociedade Brasileira de Computação, XXVII SEMISH*, julho de 2000.
- [Shibata & Katsumoto 1993] 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.
- [Silberschatz et al 1997] Silberschatz, Abraham; Korth, Henry F.; Sudarshan S. *Database System Concepts*. Third Edition, McGraw-Hill, 1997.
- [Silva et al 1999] 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
- [TaFWeb 2000] TaFWeb Software, "LinkProver 2.1b", online março/2000, disponível na internet: <http://www.tafweb.com/linkprover.html>.
- [Utting & Yankelovich 1989] Utting, K.; Yankelovich, N. "Context and Orientation in Hypermedia Networks" *ACM Trans. on Information Systems*, v.7, n.1, p.58-84, January 1989.
- [Vitali & Durand 1995] Vitali, F.; Durand, D. G., *Using versioning to support collaboration on the WWW*, Fourth World Wide Web Conference, December 1995
- [Watchfire 2000] Watchfire, "Linkbot Enterprise Server", online março/2000, disponível na internet: <http://www.watchfire.com/products/linkenterprise.htm>.
- [Whiting et al 1996] Whiting, Bill; Morgan, Bryan; Perkins, Jeff *Teach Yourself ODBC in 21 days*. Publishing SAMS, 1996.
- [Young 1990] 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.

Apêndice A - Base de dados em Paradox do DB-LiOS

Tabela 8 - Tabela de usuários

userlios						
n°	field name	type			size	key
1	id_user	integer				*
2	name	character			50	
3	address	character			50	
4	city	character			50	
5	state	character			50	
6	zip	character			20	
7	pais	character			50	
8	email	character			80	
9	occupation	integer				
10	last_date	datetime				
n°	index name	fields	cascade	unique	asc/desc	sensitive
n°	constraint name	fields			table reference	

Tabela 9 - Descrição da tabela de Sites

site						
n°	field name	type			size	key
1	id_site	autoincrement				*
2	url	character			250	
3	domain_site	integer				
4	language	integer				
5	webmaster	character			80	
6	id_user	integer				
7	last_date	datetime				
n°	index name	fields	cascade	unique	asc/desc	sensitive
1	ind_site_url	url				
n°	constraint name	fields			table reference	
1	ref_site_user	id_user			userlios	

Tabela 10 - Descrição da tabela de versão do site

version						
nº	field name	type			size	key
1	id_site	integer				*
2	id_version	integer				*
3	id_user	integer				
4	last_date	datetime				
5	model	integer				
6	link_total	integer				
7	dead_total	integer				
8	no_read_total	integer				
9	timeout_total	integer				
10	unkown_total	integer				
11	good_total	integer				
12	pages_total	integer				
13	bytes_total	float				
14	transfer	datetime				
15	all_pages	character			1	
16	link0	integer				
↓	↓	↓			↓	↓
23	link7	integer				
24	page0	integer				
↓	↓	↓			↓	↓
152	page128	integer				
nº	index name	fields	cascade	unique	asc/desc	sensitive
nº	constraint name	fields	table reference			
1	ref_version_site	id_site	site			
2	ref_version_user	id_user	userlios			

Tabela 11 - Descrição da tabela de url

url						
nº	field name	type			size	key
1	id_url	autoincrement				*
2	url	character			250	
nº	index name	fields	cascade	unique	asc/desc	sensitive
1	ind_url	url				
nº	constraint name	fields	table reference			

Tabela 12 - Descrição da tabela de código de status http

codehttp						
nº	field name	type			size	key
1	code_http	integer				*
2	description	character			50	
nº	index name	fields	cascade	unique	asc/desc	sensitive

n°	constraint name	fields	table reference

Tabela 13 - Tabela de páginas visitadas no site

pagesite						
n°	field name	type	size	key		
1	id_site	integer		*		
2	id_version	integer		*		
3	id_page	integer		*		
4	case_page	integer				
5	link_total	integer				
6	code_http	integer				
n°	index name	fields	cascade	unique	asc/desc	sensitive
n°	constraint name	Fields	table reference			
1	ref_page_version	id_site + id_version	version			
2	ref_page_url	id_page	url			

Tabela 14 - Tabela de âncoras dos links

Anchor						
n°	field name	type	size	key		
1	id_anchor	Autoincrement		*		
2	anchor	Character	250			
n°	index name	fields	Cascade	unique	asc/desc	sensitive
1	ind_anchor	anchor				
n°	constraint name	Fields	table reference			

Tabela 15 - Tabela de links das páginas do site

Link						
n°	field name	type	size	key		
1	id_site	integer		*		
2	id_version	Integer		*		
3	id_page	Integer		*		
4	id_link	Integer		*		
5	id_anchor	integer				
6	id_target	integer				
n°	index name	fields	cascade	unique	asc/desc	sensitive
1	ind_antaso	id_site + id_version + id_anchor + id_target + id_page				
2	ind_ansota	id_site + id_version + id_anchor + id_page+ id_target				

3	ind_taanso	id_site + id_version + id_target + id_anchor + id_page				
4	ind_tasoan	id_site + id_version + id_target + id_page + id_anchor				
5	ind_soanta	id_site + id_version + id_page + id_anchor + id_target				
6	ind_sotaan	id_site + id_version + id_page + id_target + id_anchor				
n°	constraint name	fields	table reference			
1	ref_link_page	id_site + id_version + id_page	pagesite			
2	ref_anchor_anchor	id_anchor	anchor			
3	ref_target_url	id_target	url			

Tabela 16 - Tabela de classificação de links

caselink						
n°	field name	type	size	key		
1	id_site	integer		*		
2	id_version	integer		*		
3	id_page	integer		*		
4	id_link	integer		*		
5	case_link	integer		*		
n°	index name	fields	cascade	unique	asc/desc	sensitive
n°	constraint name	fields	table reference			
1	ref_case_link	id_site + id_version + id_page + id_link	link			

Apêndice B - Base de dados em PostGreSQL do DB-LiOS

Cria sequencial para id_user

```
create sequence seq_id_user;
```

Cria tabela userlios

```
create table userlios (  
id_user int4 default nextval('seq_id_user') constraint key_user primary key,  
name varchar(50), address varchar(50), city varchar(50), state varchar(50), zip varchar(20),  
pais varchar(50), email varchar(80), occupation int4, last_date datetime);
```

Cria sequencial para id_site

```
create sequence seq_id_site;
```

Cria tabela site

```
create table site (  
id_site int4 default nextval('seq_id_site') constraint key_site primary key,  
url varchar(250), domain_site int4, language int4,  
webmaster varchar(80), id_user int4, last_date datetime,  
constraint ref_site_user foreign key (id_user) references userlios );
```

Cria indice do campo url para tabela site

```
create index ind_site_url on site (url);
```

Cria tabela version

```
create table version (  
id_site int4 not null, id_version int4 not null, id_user int4 not null,  
last_date datetime, model int4, link_total int4, dead_total int4,  
no_read_total int4, timeout_total int4, unkown_total int4,  
good_total int4, pages_total int4, bytes_total float, all_pages char, transfer datetime,  
link0 int4, link1 int4, link2 int4, link3 int4, link4 int4, link5 int4,  
link6 int4, link7 int4, page0 int4, page1 int4, page2 int4, page3 int4,  
page4 int4, page5 int4, page6 int4, page7 int4, page8 int4, page9 int4,  
page10 int4, page11 int4, page12 int4, page13 int4, page14 int4, page15 int4,  
page16 int4, page17 int4, page18 int4, page19 int4, page20 int4, page21 int4,  
page22 int4, page23 int4, page24 int4, page25 int4, page26 int4, page27 int4,  
page28 int4, page29 int4, page30 int4, page31 int4, page32 int4, page33 int4,  
page34 int4, page35 int4, page36 int4, page37 int4, page38 int4, page39 int4,  
page40 int4, page41 int4, page42 int4, page43 int4, page44 int4, page45 int4,  
page46 int4, page47 int4, page48 int4, page49 int4, page50 int4, page51 int4,  
page52 int4, page53 int4, page54 int4, page55 int4, page56 int4, page57 int4,  
page58 int4, page59 int4, page60 int4, page61 int4, page62 int4, page63 int4,  
page64 int4, page65 int4, page66 int4, page67 int4, page68 int4, page69 int4,  
page70 int4, page71 int4, page72 int4, page73 int4, page74 int4, page75 int4,  
page76 int4, page77 int4, page78 int4, page79 int4, page80 int4, page81 int4,  
page82 int4, page83 int4, page84 int4, page85 int4, page86 int4, page87 int4,  
page88 int4, page89 int4, page90 int4, page91 int4, page92 int4, page93 int4,  
page94 int4, page95 int4, page96 int4, page97 int4, page98 int4, page99 int4,  
page100 int4, page101 int4, page102 int4, page103 int4, page104 int4, page105 int4,  
page106 int4, page107 int4, page108 int4, page109 int4, page110 int4, page111 int4,  
page112 int4, page113 int4, page114 int4, page115 int4, page116 int4, page117 int4,  
page118 int4, page119 int4, page120 int4, page121 int4, page122 int4, page123 int4,  
page124 int4, page125 int4, page126 int4, page127 int4, page128 int4,  
constraint key_version primary key(id_site, id_version),  
constraint ref_version_site foreign key (id_site) references site,  
constraint ref_version_user foreign key (id_user) references userlios );
```

Cria seqüencial para id_url
create sequence seq_id_url;

Cria tabela url
create table url (
id_url int4 default nextval('seq_id_url') constraint key_url primary key,
url varchar(250));

Cria índice do campo url para tabela url
create index ind_url on url (url);

Cria tabela codehttp
create table codehttp (
code_http int4 not null constraint key_codehttp primary key,
description varchar(50));

Cria tabela page
create table pagesite (
id_site int4 not null, id_version int4 not null, id_page int4 not null,
case_page int4, link_total int4, code_http int4,
constraint key_page primary key(id_site, id_version, id_page),
constraint ref_page_version foreign key (id_site, id_version) references version,
constraint ref_page_url foreign key (id_page) references url);

Cria seqüencial para id_anchor
create sequence seq_id_anchor;

Cria tabela anchor
create table anchor (
id_anchor int4 default nextval('seq_id_anchor') constraint key_anchor primary key,
anchor varchar(250));

Cria índice do campo anchor para tabela anchor
create index ind_anchor on anchor (anchor);

Cria tabela link
create table link (
id_site int4 not null, id_version int4 not null, id_page int4 not null,
id_link int4 not null, id_anchor int4 not null, id_target int4 not null,
constraint key_link primary key(id_site, id_version, id_link),
constraint ref_link_page foreign key (id_site, id_version, id_page) references pagesite,
constraint ref_anchor_anchor foreign key (id_anchor) references anchor,
constraint ref_target_url foreign key (id_target) references url);

Cria índice dos campos anchor, target e source para tabela link
create index ind_antaso
on link (id_site, id_version, id_anchor, id_target, id_page);

Cria índice dos campos anchor, source e target para tabela link
create index ind_ansota on link (id_site, id_version, id_anchor, id_page, id_target);

Cria índice dos campos target, anchor e source para tabela link
create index ind_tanso on link (id_site, id_version, id_target, id_anchor, id_page);

Cria índice dos campos target, source e anchor para tabela link
create index ind_tasoan on link (id_site, id_version, id_target, id_page, id_anchor);

Cria índice dos campos source, anchor e target para tabela link

```
create index ind_soanta on link ( id_site, id_version, id_page, id_anchor, id_target );
```

Cria índice dos campos source, target e anchor para tabela *link*

```
create index ind_sotaan on link ( id_site, id_version, id_page, id_target, id_anchor );
```

Cria tabela caselink

```
create table caselink (
```

```
id_site int4 not null, id_version int4 not null, id_page int4 not null,
```

```
id_link int4 not null, case_link int4 not null,
```

```
constraint key_class primary key(id_site, id_version, id_page, id_link, case_link),
```

```
constraint ref_case_link foreign key (id_site, id_version, id_page , id_link) references link );
```

Apêndice C - Códigos de status de HTTP

Tabela 17 - Tabela de códigos de status http

Cód	Status	Descrição
-xxx	Extended for DB-LIOS	Extendido para DB-Lios
-2	unkown codehttp	Código http desconhecido
-1	dblios timeout	Espirou tempo de espera do DB-LIOS
0	no read page	página não lida
1xx	Informational	Informacional
100	Continue	Continuar
101	Switching Protocols	Mudança de protocolo
2xx	Successful	Sucesso
200	OK	OK
201	Created	Criado
202	Accepted	Aceito
203	Non-Authoritative Information	Informação sem necessidade de autorização
204	No Content	Sem conteúdo
205	Reset Content	Conteúdo resetado
206	Partial Content	Conteúdo parcial
3xx	Redirection	Redireção
300	Multiple Choices	Múltiplas escolhas
301	Moved Permanently	Movido Permanentemente
302	Found	Encontrado
303	See Other	Ver Outro
304	Not Modified	Não Modificado
305	Use Proxy	Uso do Proxy
306	(Unused)	(Código não utilizado)
307	Temporary Redirect	Redirecionamento temporário
4xx	Client Error	Erro do Cliente
400	Bad Request	Requisição mal elaborada
401	Unauthorized	Não autorizado
402	Payment Required	Requerido pagamento
403	Forbidden	Acesso Proibido
404	Not Found	Não encontrado
405	Method Not Allowed	Método de acesso não disponível
406	Not Acceptable	Não aceitável
407	Proxy Authentication Required	Requerido autenticação do Proxy
408	Request Timeout	Timeout da Requisição
409	Conflict	Conflito
410	Gone	Perdido
411	Length Required	Tamanho requerido
412	Precondition Failed	Falha na Pré-condição
413	Request Entity Too Large	Requisição da entidade muito longa
414	Request-URI Too Long	URI requisitada muito longa
415	Unsupported Media Type	Tipo de Mídia não suportada
416	Requested Range Not Satisfiable	Faixa requerida não satisfeita

417	Expectation Failed	Falha na expectativa
5xx	Server Error	Erro do Servidor
500	Internal Server Error	Erro interno no Servidor
501	Not Implemented	Não implementado
502	Bad Gateway	Gateway Inválido
503	Service Unavailable	Serviço não disponível
504	Gateway Timeout	Timeout no Gateway
505	HTTP Version Not Supported	Versão do HTTP não suportado