

**GECON: Um Gerenciador de Conteúdos na Web para
não especialistas em informática**

Roberto Romani

Trabalho Final de Mestrado Profissional

UNICAMP
BIBLIOTECA CENTRAL
SEÇÃO CIRCULANTE

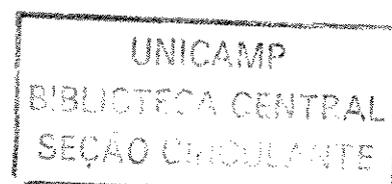
GECON: Um GERenciador de CONteúdos na Web para não especialistas em informática

Roberto Romani

Fevereiro de 2004

Banca Examinadora:

- Profa. Dra. Maria Cecília Calani Baranauskas (Orientadora)
- Prof. Dr. José Oscar Fontanini de Carvalho
CEATEC Centro de Ciências Exatas Ambientais e de Tecnologia – PUC-
Campinas
- Prof. Dra. Anamaria Gomide
Instituto de Computação - Unicamp
- Prof. Dra. Maria de Fátima R. O. Pires Silva
Coordenadoria Geral de Informática – Unicamp



UNIDADE	BC
Nº CHAMADA	+UNICAMP
	R661g
V	EX
TOMBO BC/	63109
PROC.	16.86.05
C	<input type="checkbox"/>
	D <input checked="" type="checkbox"/>
PREÇO	1,00
DATA	31-03-05
Nº CPD	

BIBID - 346076

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Romani, Roberto

R661g GECON: Um gerenciador de conteúdos na Web para não especialistas em informática / Roberto Romani -- Campinas, [S.P. : s.n.], 2004.

Orientador : Maria Cecília Calani Baranauskas

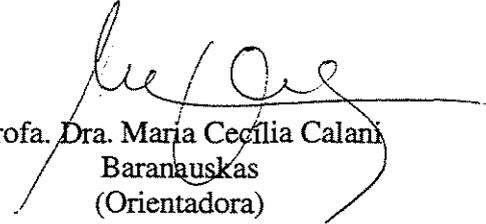
Trabalho final (mestrado profissional) - Universidade Estadual de Campinas, Instituto de Computação.

1. Interação homem-máquina. 2. Educação à distância.
3. Gerenciamento da informação. I. Baranauskas, Maria Cecília Calani.
II. Universidade Estadual de Campinas. Instituto de Computação. III.
Título.

GECON: Um GEreenciador de CONteúdos na Web para não especialistas em informática

Este exemplar corresponde à redação final do Trabalho Final devidamente corrigida e defendida por Roberto Romani e aprovada pela Banca Examinadora.

Campinas, 19 de fevereiro de 2004.



Prof. Dra. Maria Cecília Calani
Baranauskas
(Orientadora)

Trabalho Final apresentado ao Instituto de Computação, Unicamp, como requisito parcial para a obtenção do título de Mestre em Computação na área de Engenharia de Computação.

TERMO DE APROVAÇÃO

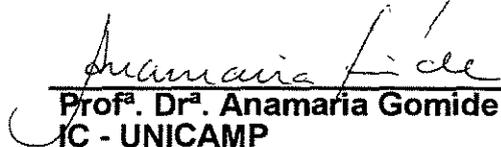
Tese defendida e aprovada em 19 de fevereiro de 2004, pela Banca Examinadora composta pelos Professores Doutores:



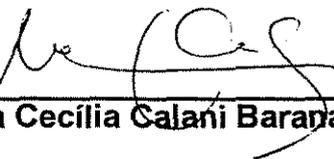
Prof. Dr. José Osear Fontanini de Carvalho
Instituto de Informática – PUC-Campinas



Profª. Drª. Maria de Fátima Ridolfi Ordini Pires da Silva
Comitê Gestor de Informática - UNICAMP



Profª. Drª. Anamaria Gomide
IC - UNICAMP



Profª. Drª. Maria Cecília Calani Baranauskas
IC - UNICAMP

A minha esposa Luciana, aos meus pais e em especial ao meu filho Lucas.

Agradecimentos

A minha esposa Luciana pelo apoio, incentivo, compreensão, carinho, paciência e muito mais.

Aos meus pais, sem eles esse trabalho não poderia ser escrito.

Em especial ao meu filho Lucas pelas muitas horas de atenção que lhe foram desviadas.

À Profª Cecília Baranauskas pela orientação, confiança e incentivo na realização desta pesquisa.

A toda a equipe de EAD da Unicamp pela colaboração, em especial aos estagiários que participaram da implementação.

Aos colegas do Mestrado Profissional pelo companheirismo.

Ao grupo de orientados da prof. Cecília pelas sugestões.

Aos alunos do curso de especialização da PUC-Campinas do 2º semestre de 2002 pela realização das avaliações heurísticas.

Resumo

Ao longo dos anos, a Internet tem sido utilizada como meio de comunicação, divulgação, repositório de informação e, mais recentemente, para *e-learning*. Todos estes usos basicamente ocorrem por meio de sites, um conjunto de páginas estáticas ou dinâmicas, organizadas em uma estrutura chamada de *World Wide Web*. Desta forma, a web assume um papel crucial na sociedade da informação em que vivemos pois é um meio de comunicação onde qualquer pessoa pode se expressar por meio de um *site*. Entretanto aqueles que não detêm o conhecimento técnico para criar e manter um *site* dependem obrigatoriamente de terceiros.

Os gerenciadores de conteúdo são sistemas computacionais que dinamizaram e facilitaram a tarefa de criar e manter um *site*. Contudo sua complexidade é tal que seu uso se mantém restrito a usuários com conhecimentos avançados em informática.

O objetivo deste trabalho é estudar aspectos de usabilidade e propor um gerenciador de conteúdos – GECON - que possa ser manipulado por pessoas com conhecimentos básicos em informática, estendendo a esse grupo a possibilidade de criar e manter um *site* sem a dependência de especialistas na área. Inspeções de usabilidade realizadas durante o desenvolvimento do GECON mostraram-se um método eficaz para o levantamento de problemas de usabilidade, que puderam ser tratados em iterações seguintes do ciclo de desenvolvimento. O teste piloto realizado com os primeiros usuários do GECON mostrou que o sistema é eficiente e tem potencial para permitir o uso por pessoas com poucos conhecimentos da tecnologia web.

Abstract

Since early years, Internet has been used as a communication and diffusion medium, as an information repository and more recently as a tool for e-learning. Sites, a set of dynamic or static pages, are used to allow these uses. A World Wide Web is the framework in which these pages exist. Web assumes a crucial role in the information society where we live. The Web is a communication medium through which everyone can express himself/herself using a site. However, people who do not get technical knowledge to create or to maintain a site, obligatorily depends on another ones.

Content management systems (CMS) are computational systems that improve and facilitate the task of creating and maintaining a site. Nevertheless its complexity is so high that its use is restrict to users with advanced knowledge in Informatics.

The objective of this work is to study usability aspects and to propose a new CMS – GECON - directed to people with basic knowledge in Informatics. This CMS proposal is to allow groups of people to create and maintain a site without help of experts. Usability inspections carried during the development of GECON showed to be an effective method to identify usability problems. These problems were corrected in the following iterations of the development life cycle. Pilot tests accomplished with first users of GECON showed that the system is efficient and has potential to be used by people with short knowledge in the web technology.

Sumário

Resumo	ix
Abstract	x
Capítulo 1 – Introdução	19
Capítulo 2 - Sistemas de Gerenciamento de Conteúdo	25
2.1 O design e a avaliação de Interfaces	27
2.2 A importância da usabilidade na web	29
2.3 Definição e estrutura de um CMS	31
2.4 O PHP-Nuke	33
2.5 O Plone	37
2.6 Porquê desenvolver outro CMS?	42
2.7 Considerações finais	43
Capítulo 3 -GECON – Gerenciador de CONteudos: Design e Desenvolvimento	45
3.1 Aspecto funcional	46
Administração	48
A opção Geral	50
Elemento Logo	51
Elemento Banner	51

Elemento Menu Horizontal	53
Elemento Menu Vertical	54
Elemento Área Principal	56
Elemento Parcerias	58
Elemento Rodapé	58
3.2 Aspecto Arquitetônico	59
Estrutura	61
Pasta administração	62
Pasta assistentes	63
Pasta CSS	64
Pasta imagens	64
Pasta js	64
Pasta sites	65
3.3 Considerações finais	67
Capítulo 4 Análise da Interface do GECON: inspeção e teste	69
4.1 Avaliação Heurística	71
4.1.1 Descrição do método	71
4.1.2 Contexto da aplicação da avaliação heurística	74
4.1.3 Resultados das avaliações	74
Heurística 1: Visibilidade do status do sistema	75

Heurística 2: Coerência do sistema com o mundo real	77
Heurística 3: Controle e liberdade	78
Heurística 4: Consistência e padronização	79
Heurística 5: Prevenção de erros	80
Heurística 6: Reconhecer é melhor que relembrar	81
Heurística 7: Flexibilidade e eficiência de uso	82
Heurística 8: Estética e design suficiente	83
Heurística 9: Auxilia no reconhecimento, diagnóstico e recuperação de erros?	84
Heurística 10: Ajuda e documentação	85
4.1.4 Discussão geral sobre os resultados da Avaliação Heurística	86
4.2 Avaliação Piloto de Uso do Sistema	91
4.2.1 Contexto da avaliação piloto	91
4.2.2 Resultados da avaliação piloto	93
Trabalho com o assistente de eventos	94
Trabalho com o assistente de publicações	98
Trabalho com o assistente de links	101
Trabalho com o assistente de HTML	101
4.2.3 Conclusão sobre a avaliação piloto de uso	105
4.3 Considerações finais	105

Capítulo 5 – Conclusões	107
Referências	113
Anexo	117

Lista de Figuras

Figura 2.1 - Tela com o menu administrativo de PHP-Nuke	36
Figura 2.2 – Tela de administração dos Blocos do PHP-Nuke	37
Figura 2.3 – Tela com a pasta de um usuário membro do Plone	41
Figura 2.4 – Tela de edição de um conteúdo recém criado no Plone	41
Figura 2.5 – Tela que permite alterar o estado do conteúdo no Plone	42
Figura 3.1 – Elementos do GECON divididos por regiões	47
Figura 3.2 – Interface administrativa do GECON	48
Figura 3.3 -Janela administrativa da opção Geral	49
Figura 3.4 – Janela administrativa do elemento Banner	52
Figura 3.5 – Janela administrativa do Menu Horizontal	53
Figura 3.6 – Janela administrativa do Menu Vertical	55
Figura 3.7 – Assistente de HTML	57
Figura 3.8 – GECON entre o administrador do <i>site</i> e o Zope	60
Figura 3.9 – Estrutura hierárquica do GECON	61
Figura 3.10 – Estrutura hierárquica expandida	66

Figura 4.1 – Gráfico da Heurística Visibilidade	76
Figura 4.2 – Gráfico da Heurística Coerência com o mundo real	77
Figura 4.3 - Gráfico da Heurística Controle e Liberdade	78
Figura 4.4 - Gráfico da Heurística Consistência e Padronização	79
Figura 4.5 - Gráfico da Heurística Prevenção de Erro	80
Figura 4.6 - Gráfico da Heurística Reconhecer X Lembrar	81
Figura 4.7 - Gráfico da Heurística Flexibilidade e Eficiência de Uso	82
Figura 4.8- Botões antes da avaliação heurística	83
Figura 4.9 - Botões após a redefinição	83
Figura 4.10 - Gráfico da Heurística Estética e Design Suficiente	84
Figura 4.11 - Gráfico da Heurística Recuperação de Erros	85
Figura 4.12 - Gráfico da Heurística Ajuda e Documentação	86
Figura 4.13 - Tela de inclusão do elemento parcerias	88
Figura 4.14 - Tela de inclusão do elemento parcerias remodelada	89
Figura 4.15 - Gráfico dos Problemas por Heurística	90
Figura 4.16 - Site de EAD feito com GECON	92

Figura 4.17 - Interface administrativa com texto explicativo junto aos ícones	94
Figura 4.18 - Formulário do assistente de eventos	95
Figura 4.19 - Listagem dos eventos da categoria Unicamp	96
Figura 4.20 - Página gerada a partir do campo ementa	97
Figura 4.21 - Tela do assistente de <i>links</i> antes da separação do formulário de categorias	99
Figura 4.22 - Tela de assistente com submenu para incluir categoria	100
Figura 4.23 - Tela inicial para selecionar assistente	102
Figura 4.24 - Tela com submenu para selecionar assistente	103
Figura 4.25 - Tela com estrutura hierárquica navegável contendo as categorias e as páginas de conteúdo com um menu <i>popup</i>	104

Capítulo 1

Introdução

Uma das metáforas mais usadas para descrever a web é “teia”, porque possui uma série de pontos interconectados sem uma ordem definida, lembrando uma teia de aranha. Estes pontos são, na verdade, computadores - servidores - com um ou mais *sites* repletos de informação. A web, então, pode ser considerada um grande repositório de informação sem organização pré-estabelecida. Devido a esta característica muito material, tenha ou não qualidade, tem sido colocado disponível na web. Cabe ao usuário classificar este conteúdo no momento da sua pesquisa.

É certo que, como veículo rápido e eficiente de acesso à informação, a web tem se destacado, fornecendo suporte para uma série de propósitos nos últimos anos. Desde seu início, a comunidade ligada à educação percebeu esse potencial da Internet e o tem explorado. Os recursos da web podem ser usados com um propósito educacional dada a facilidade de disponibilizar material didático pedagógico ao público em geral ou a grupos restritos, por meio de *sites* educacionais. No entanto, as ferramentas disponíveis para criação e manutenção de *sites* podem transformar-se em barreiras para educadores que não necessariamente detêm conhecimentos técnicos para usá-las.

Ferramentas como o *Dreamweaver da Macromidia* (Macromidia, 2003) ou o *Frontpage da Microsoft* (Microsoft, 2003), além de caras, exigem que um usuário leigo faça um curso para conseguir utilizá-las tamanha a sua variedade de recursos. Muitos deles dificilmente são usados na maioria dos *sites* criados. Além disso, os arquivos gerados por estas ferramentas requerem o uso de softwares de FTP (File Transference Protocol) para publicação do *site*. Normalmente, essas ferramentas ficam instaladas em máquinas diferentes daquelas que hospedam os *sites*, sendo necessário fazer a transferência dos arquivos.

Essas ferramentas manuais para criação de *sites* geram um conjunto de páginas em formato HTML (HyperText Markup Language) que são trabalhosas para serem atualizadas. Com isso, muitos *sites* passam anos com a mesma interface e conteúdo.

No tocante a educação, a dificuldade de atualização constante de conteúdo associada ao desconhecimento técnico em informática, levou a uma quantidade relativamente pequena de material didático disponível para acesso via web. No entanto, é importante destacar a iniciativa de vários grupos no Brasil e no exterior que criaram ambientes de autoria para educação. Devido às características destes ambientes, eles têm sido mais utilizados para educação a distância. Vários cursos têm sido criados com estas ferramentas, mas atingem um público restrito já que pressupõem o acesso através de senhas.

Com o avanço tecnológico, surge uma nova possibilidade de tornar material disponível na web de forma mais facilitada. Esta tecnologia conhecida como sistemas gerenciadores de conteúdo ou CMS (do inglês, *Content Management System*) possibilita a criação e gerenciamento de conteúdos. Estes sistemas eliminam a etapa de transferência de arquivos, permitindo a manipulação das páginas do *site* diretamente no servidor. Entretanto, ainda são projetados para um público com conhecimentos em informática pois usam muitas vezes termos técnicos e uma interface voltada a usuários experientes.

Este cenário de necessidade de criação de *sites* para publicação de material educacional associado à falta de ferramentas com interface de usuário e funcionamento dirigido para público não sofisticado em informática era a realidade do grupo de Educação a Distância (EAD) da Universidade Estadual de Campinas (Unicamp) no ano de 2002.

A equipe de EAD foi criada pela Unicamp com o objetivo de fomentar e apoiar o uso de EAD dentro da universidade. Uma das demandas atendidas por essa equipe é prover

informações atualizadas sobre EAD à comunidade acadêmica. Um dos canais de comunicação entre a equipe e a comunidade é o *site* de EAD da Unicamp hospedado no endereço <http://www.ead.unicamp.br>.

Um exemplo da dificuldade de gerenciamento de *sites* como é relatado neste trabalho, era a do próprio *site* do grupo de EAD. Até 2002, este *site* era mantido manualmente. Quase todas as páginas eram feitas em HTML estático, o que tornava a manutenção bastante difícil. Para acrescentar um item no menu principal, por exemplo, era preciso alterar todas as páginas pois o menu era replicado estaticamente em cada uma delas.

Além disso, várias pessoas são responsáveis pela atualização do conteúdo das páginas e no *site* antigo, elas não tinham a permissão necessária para editar um arquivo HTML. Este fato tornava alterações simples, como corrigir a grafia de uma palavra, uma tarefa trabalhosa que envolvia duas ou mais pessoas por um tempo maior que o necessário.

Outra atividade da equipe é atender às solicitações de professores e grupos de pesquisa ligados a EAD para ajudar na elaboração, manutenção e hospedagem de *sites* de projetos ligados a EAD.

Tomando por base o tempo gasto com a manutenção do *site* da equipe concluiu-se que para atender adequadamente essa crescente demanda seria necessário uma outra solução tecnológica. Solução esta que desse condições às pessoas de manterem seus *sites* de forma independente. Desta forma, a equipe teria apenas o trabalho de criação inicial, hospedagem e eventual suporte técnico.

Em busca desta solução, vários sistemas gerenciadores de conteúdo foram analisados. Entretanto, todos os CMSs pesquisados mostraram-se inadequados, quer seja pela complexidade de operação, dificuldade de manuseio da interface ou por problemas de

segurança. Decidiu-se então projetar e implementar um sistema de gerenciamento de conteúdos que atendesse tanto as necessidades internas da equipe quanto as externas de hospedagem e suporte de *sites* ligados a EAD dentro da Unicamp.

O objetivo central deste trabalho é propor e implementar um CMS (o GECON) dirigido ao público não especialista em informática, apresentando e descrevendo todas as etapas de seu desenvolvimento.

Projetar uma interface visando boa usabilidade e validar o sistema tanto por pessoas com conhecimentos em design de interfaces quanto pela própria equipe de EAD, também são objetivos a serem alcançados antes da sua implantação.

Integrantes da equipe de EAD com formação em pedagogia foram considerados representantes de usuários não especialistas em computação que validaram o GECON. Essa validação pode ser considerada um primeiro teste do sistema por não especialistas, entretanto, é necessário que o GECON seja validado por um grupo maior de não especialistas em informática o que será feito *a posteriori* não estando incluso no escopo desse trabalho.

No capítulo 2 desta dissertação são descritos com mais detalhes os sistemas gerenciadores de conteúdo e suas características básicas. São descritos e analisados dois dos CMSs mais utilizados, o Plone e o PHP-Nuke.

No capítulo 3 é descrito o sistema GECON, suas funcionalidades e formas de operação. Além disso, telas são utilizadas para ilustrar detalhes relevantes do sistema.

No capítulo 4 é apresentada uma avaliação heurística de sua versão inicial, que envolveu cerca de 14 grupos formados por pessoas com alguma experiência em análise de interação humano-computador. Neste capítulo, também são discutidos resultados de um

teste piloto realizado pela equipe de EAD para validação do sistema antes de colocá-lo em uso.

Finalmente, no capítulo 5 são enumeradas as conclusões e apontadas possibilidades de trabalhos futuros.

Capítulo 2

Sistemas de Gerenciamento de Conteúdo: visão geral

Introdução

A evolução das interfaces acompanha o desenvolvimento de novas tecnologias de hardware. Inicialmente, a interação do ser humano com a máquina se dava por meio do acionamento de válvulas e transistores, passando pelo uso de menus e linhas de comando até chegar à manipulação direta. Com a evolução da internet um novo paradigma de interface surge, a chamada interação humano-humano (Ishii, 1993), por meio de ferramentas de comunicação e *sites*.

Um *site* é um conjunto de páginas inter relacionadas escritas em HTML, uma linguagem de marcação de texto que é interpretada pelos navegadores web. Eles ficam hospedados em computadores com servidores web e podem ser acessados de qualquer parte do mundo por meio da rede mundial de computadores: a Internet.

No início da web as páginas HTML dos *sites* eram escritas uma a uma em editores de texto convencionais. Essa tarefa demandava um grande esforço na criação e principalmente na manutenção dos *sites*. Posteriormente, foram desenvolvidos editores que permitem criar páginas HTML visualmente, com interface gráfica do tipo WYSWYG (*what you see is what you get*).

Com a evolução da web, muitos *sites* começaram a usar páginas dinâmicas que são montadas no momento em que são acessadas apresentando informações recuperadas automaticamente de bancos de dados. Essa evolução tecnológica tornou muito mais complexo o gerenciamento dos *sites*, o que levou ao desenvolvimento de sistemas com o propósito de facilitar essa tarefa. Essas ferramentas, também chamadas de CMS, suportam

a criação, gerenciamento, distribuição e publicação de conteúdo na Internet (Pokorny, 2001; Robertson, 2003). Além disso, atualmente os usuários têm mais domínio sobre a tecnologia e conseqüentemente esperam muito mais dos *sites*. Um novo aspecto passa a ser de extrema importância – a usabilidade de *sites* na web.

Neste capítulo é feita uma explanação sobre design e avaliação de interfaces, enfatizando a importância da engenharia de usabilidade na criação da interface de sistemas CMSs, seguida de uma descrição dos CMSs e uma análise geral de dois CMS que mais se aproximaram das necessidades discutidas na capítulo 1.

2.1 O design e a avaliação de Interfaces

Interface, de forma geral, é uma superfície de contato que reflete as propriedades físicas daqueles que interagem. Por exemplo: uma maçaneta é a interface entre uma pessoa e a porta. A interface (maçaneta) será mais adequada dependendo de quão bem projetada for para a pessoa que vai utilizá-la. Quando o conceito de interface surgiu algumas décadas atrás, entendia-se Interface como o hardware e o software através do qual o homem e o computador podem se comunicar. Atualmente, o conceito evoluiu e também são considerados aspectos cognitivos e emocionais das experiências do usuário (Laurel, 1990). Dessa forma, entende-se a interface como algo que se pode mapear, fazer o design, projetar, implementar e juntar à funcionalidade do sistema no desenvolvimento do software.

Como a tecnologia de computação tem se tornado disponível para mais e mais pessoas com uma variedade grande de equipamentos e contextos, a necessidade de ser mais acessível, fácil de usar e ter um melhor engajamento com o usuário aumentou

drasticamente. Como a natureza do problema tem sido explorada e articulada com mais detalhamento, pessoas com diferentes níveis de conhecimento e de posicionamento têm se envolvido nesse processo. O design de interface ganhou atenção e preocupação maiores por parte dos desenvolvedores e muitos aspectos anteriormente ignorados são agora considerados. Os *designers* se preocupam com o usuário da interface, com suas atitudes, tarefas, como funciona sua memória, percepção entre outras características.

Mas, o impacto dessa tecnologia por si só não é suficiente para tornar os artefatos baseados no computador - em especial o *software* - mais usáveis. Aplicações criativas da tecnologia que estendem o “poder” do humano têm favorecido o fator de usabilidade - potencial para ser usável pelo humano – à medida que nossas máquinas tornam-se mais sofisticadas.(Dix et al. 1993).

A interação entre o usuário e o computador é afetada tanto pela tecnologia quanto por fatores sociais e organizacionais do contexto em que o usuário está inserido. Em relação aos aspectos tecnológicos, paradigmas de interação têm surgido ao longo da própria história da área de Interação Humano-Computador (IHC) e do desenvolvimento de sistemas interativos: interfaces WIMP (*windows, icons, menus e pointers*), manipulação direta, hipertexto, etc. Em relação aos fatores humanos, a literatura (Norman, 1986; Laurel, 1990; Winograd, 1996; Nardi, 1995) tem apresentado princípios para interação humano-computador que dependem menos da tecnologia e muito mais de um entendimento profundo do elemento humano na interação. Os paradigmas de interação, se considerados isoladamente não evidenciam como se dá o suporte ao usuário na realização de suas tarefas. Os princípios de interação podem ser definições gerais e abstratas não muito úteis ao

designer. Por outro lado, a consciência dos fatores humanos pode ajudar a limitar os efeitos negativos do uso da tecnologia na interação.

Apesar da problemática discutida, é senso comum o fato de que atualmente são construídos sistemas mais usáveis. No entanto, ainda há muito a se conhecer e fazer em termos de *design* de sistemas interativos. Dix et al. (1993) afirmam que o objetivo primeiro de um sistema interativo é permitir ao usuário alcançar metas particulares em algum domínio de aplicação; isto é, o sistema interativo deve ser “usável”.

Na criação de objetos para uso das pessoas, algumas decisões são tomadas e tais objetos carregam uma intenção do *designer*, de percepção e uso, para com o usuário final. A Engenharia de Usabilidade (Nielsen, 1993) propõe o uso de metas de usabilidade no processo de *design*. Os princípios básicos de usabilidade envolvem três categorias principais: facilidade com que novos usuários podem efetivamente começar a interagir; diversidade de maneiras com que o usuário e o sistema trocam informação e o nível de suporte que o usuário tem para determinar seu sucesso e a avaliação de suas metas. Nielsen e Molich (1990) propuseram um método conhecido como “avaliação heurística”, em resposta à necessidade de efetivamente se avaliar o design dos sistemas. A avaliação heurística possui um critério específico para guiar a avaliação. A abordagem é simples e relativamente rápida, embora um certo conhecimento e número de avaliadores sejam necessários para a aplicação das heurísticas de usabilidade.

2.2 A importância da usabilidade na web

Muitas definições sobre o que é usabilidade têm sido propostas na literatura. Por

exemplo, o pesquisador francês Dominique Scapin (Scapin, 1990) considera que a usabilidade está diretamente ligada ao diálogo na interface e é a capacidade do software de permitir que o usuário alcance suas metas de interação com o sistema. Para Jakob Nielsen (Nielsen, 1993), três são os aspectos fundamentais para a percepção de uma boa usabilidade por parte do usuário: ser de fácil aprendizagem, permitir utilização eficiente e apresentar poucos erros.

É fundamental para qualquer sistema que a interface tenha uma boa usabilidade, porém, em aplicações web isso se torna especialmente importante. Na web, os usuários experimentam a usabilidade dos *sites* antes de adquirir o produto. Se eles não encontram o produto que desejam, podem facilmente procurá-lo no concorrente que está a um clique do *mouse*, diferentemente dos softwares de prateleira que são adquiridos primeiro e a usabilidade é verificada depois. Ou seja, se o usuário não encontra a informação que procura porque não entende como navegar naquele *site*, será muito mais simples para ele buscar a informação em um *site* onde seu esforço cognitivo seja menor.

Na web, o tempo que o usuário dispõe para aprender a interagir com a interface é nenhum. Ou seja, se ele conseguir navegar e encontrar as informações que procura, ótimo; ele e a interface atingiram seu objetivo. Caso a navegação não seja de fácil compreensão, ele certamente não encontrará o que procura indo imediatamente para outros *sites*.

A maioria dos CMSs não tem uma preocupação específica quanto à usabilidade dos *sites* construídos por eles. Esse cuidado é transferido para a pessoa responsável pela construção do *site*. Não é possível que a ferramenta CMS garanta que o *site* gerado tenha uma boa usabilidade pois esta depende em grande parte da arquitetura da informação no *site*, e isso não pode ser pré-estabelecido pela ferramenta ficando a cargo do responsável

pela construção do site. Porém, seria possível evitar erros básicos simplesmente padronizando a navegação no *site*, o que pode ser feito ou no mínimo sugerido pela ferramenta.

2.3 Definição e estrutura de um CMS

Atualmente, os usuários da web têm expectativas maiores em relação ao conteúdo dos *sites* e ao tempo de atualização dos mesmos. Para que os usuários retornem com frequência a um site, ele deve oferecer informações de qualidade e com atualizações frequentes. Manter um *site* sob essas condições pode se tornar uma tarefa difícil para uma única pessoa, principalmente para *sites* que disponibilizam informação em tempo real, como por exemplo portais de conteúdo como Yahoo, portais de previsão do tempo, entre outros.

Para fornecer conteúdo atualizado de forma rápida e com qualidade aos usuários finais, os *websites* precisaram adotar um novo paradigma de desenvolvimento. A administração das antigas páginas estáticas dos *sites* na web desenvolvidas em HTML está gradativamente sendo substituída por CMSs. A grande vantagem de um CMS consiste em permitir que várias pessoas possam dividir essa tarefa sem criar inconsistências na estrutura e no design do *site*.

Existem no mercado atualmente vários CMSs, dentre eles o OpenCMS (Open-CMS, 2003) desenvolvido na linguagem de programação Java (Java 2003) e que faz uso da tecnologia XML (eXtended Markup Language), o PHP-Nuke (PHP-Nuke, 2003) desenvolvido em PHP (Hypertext Preprocessor), uma linguagem de programação interpretada, que é especialmente interessante em desenvolvimento para a Web, a principal

característica do PHP-Nuke é sua modularidade, o PostNuke (PostNuke, 2003) também foi desenvolvido em PHP e é modular, ele é voltado para construção de comunidades virtuais. O Squishdot (Squishdot, 2003) foi desenvolvido em Zope (Zope 2003) uma plataforma de desenvolvimento de aplicações web. O Squishdot é voltado para sites de notícias permitindo a publicação de pequenos artigos, notícias, anuncios etc. O Slach (Slach 2003) foi desenvolvido na linguagem Perl e também é dirigido para sites de notícias. O Cocoon é um projeto da Apache Software Foundation (Apache 2003) escrito em Java, usa tecnologia XML e possui subprojetos como o Lenya (Lenya 2003) e o Wyona (Wyona 2003). O WebMake (WebMake, 2003) desenvolvido em Perl também faz uso da tecnologia XML. O Xoops (Xoops, 2003) desenvolvido em PHP é baseado em orientação a objetos. O Plone (Plone 2003), desenvolvido em Zope, além de usar conceitos de orientação a objetos, é fortemente baseado em um workflow para publicação de conteúdo. CMS são desenvolvidos em diversas linguagens com diferentes enfoques; mas, as suas funções básicas são: criar, gerenciar, distribuir e publicar conteúdo na web.

Um CMS deve cobrir todo o ciclo de vida das páginas de um *site* desde sua criação até sua publicação. Também deve fornecer ferramentas para gerenciamento da estrutura do *site*, da aparência das páginas publicadas e da navegação para os usuários finais.

Pode-se subdividir a funcionalidade de um CMS em 4 categorias:

- criação do conteúdo
- gerenciamento do conteúdo
- publicação
- apresentação

A criação e a atualização do conteúdo são feitas por meio de um ambiente de autoria

com facilidades de edição, permitindo que essa tarefa seja realizada por pessoas leigas em tecnologia da informação e computação. Em geral os CMSs são aplicações web que possibilitam a atualização do conteúdo remotamente. Estas alterações são feitas utilizando-se navegadores web (Netscape, Mozilla, Internet Explorer, etc.), o que elimina a necessidade de acesso direto ao servidor, no qual encontra-se instalado o *site*, por meio de programas de FTP, SSH, Telnet ou outros.

O conteúdo fica armazenado em um repositório, em geral um banco de dados, permitindo assim seu gerenciamento e possibilitando inclusive manter um histórico de cada conteúdo.

Uma vez que o conteúdo esteja no repositório, ele pode ser publicado. A maioria dos CMSs possui um ou mais modelos (*templates*) de páginas que são aplicados ao conteúdo para gerar a página HTML no momento em que alguém acessa a página. Dessa forma é possível alterar o *layout* da página sem reprogramação.

2.4 O PHP-Nuke

O PHP-Nuke (PHP-Nuke, 2003) foi desenvolvido originalmente pelo canadense radicado na Venezuela Francisco Burzi. Foi um dos primeiros CMSs desenvolvidos em PHP, uma linguagem flexível criada para desenvolver aplicações web, e foi baseado em um sistema de publicação de notícias escrito por ele na linguagem Perl chamado Nuke (Broder, 2003).

Esse sistema tornou-se muito popular no final de 2000. Passou a contar com a colaboração de integrantes da comunidade de software livre e em 2001 obteve apoio financeiro da Mandrake Software (Mandrake, 2003), distribuidora do Mandrake Linux.

O PHP-Nuke pode ser usado como um gerenciador de artigos, no qual podem ser registrados autores, revisores e moderadores, criando uma comunidade que alimenta o *site* com informações novas a qualquer momento.

Outras funcionalidades foram incorporadas aos CMSs ao longo da sua história, como por exemplo o gerenciador de *templates* (*layout* ou design da página de web). Isso permite que o administrador do CMS, com um certo conhecimento em HTML, possa personalizar sua página de modo prático, sem a necessidade de alteração no código fonte ou seja, nos *scripts* dinâmicos do CMS.

A principal característica do PHP-Nuke é sua estrutura modular, que tem como princípio uma arquitetura parecida com “blocos de encaixe”. Essa arquitetura faz com que o PHP-Nuke adeque-se a diferentes situações, permitindo a integração de novas funcionalidades facilmente. Isto possibilita que novos recursos sejam desenvolvidos, não somente pela equipe principal mas também por colaboradores ou pelo próprio administrador do PHP-Nuke. No entanto, é necessário que o desenvolvedor tenha um conhecimento avançado na linguagem PHP.

A instalação do PHP-Nuke requer bons conhecimentos técnicos em informática pois antes de instalar o gerenciador é preciso instalar e configurar o banco de dados, o servidor WEB e o interpretador da linguagem PHP. Após a instalação do banco é preciso que os *scripts* SQL para a criação de bancos e tabelas sejam interpretados e por último é necessário preencher corretamente o arquivo de configuração que normalmente pede o tipo do banco de dados, o nome do banco de dados, o usuário e a senha do banco de dados e o endereço (*path*) completo do servidor.

O PHP-Nuke pode ser utilizado com vários bancos de dados diferentes, a saber

MYSQL(MYSQL, 2003), MS SQL (MS SQL, 2003), ORACLE(ORACLE, 2003), POSTGRE-SQL (POSTGRE-SQL, 2003), DB2 (DB2, 2003), ACCESS (ACCESS, 2003).

O PHP-Nuke possui um gerenciador de *templates* que tem uma estrutura simétrica, ou seja, elementos colocados no lado esquerdo do *site* precisam ter o mesmo tamanho dos colocados no lado direito. O sistema possui um conjunto de *templates* pré-definidos que podem ser utilizados no *site*.

Algumas “brechas” que permitem a invasão do *site* construído em PHP-Nuke são regularmente descobertas e corrigidas. Estas versões atualizadas ficam disponíveis no *site* do PHP-Nuke e devem ser “baixadas” e atualizadas localmente para impedir possíveis ataques.

As Figuras 2.1 e 2.2, a seguir, ilustram instantâneos de tela do referido sistema.

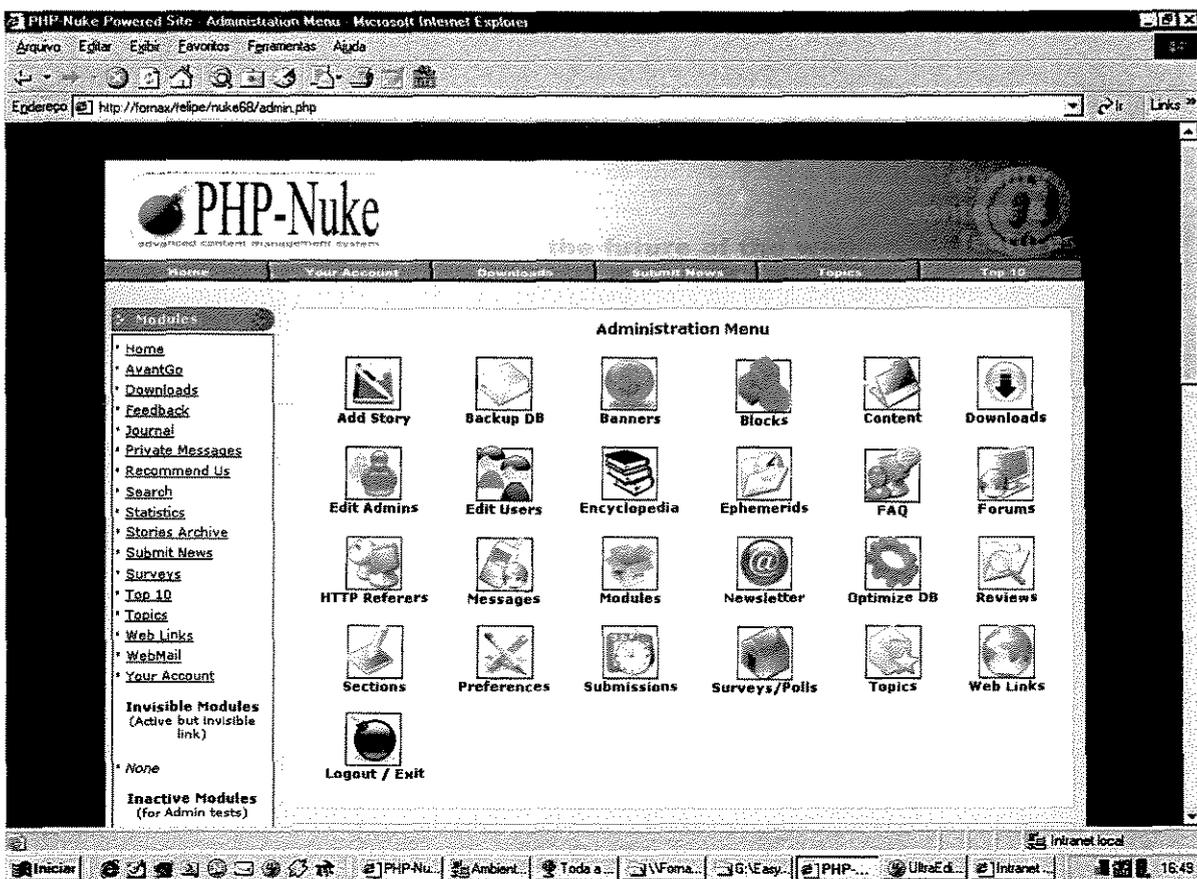


Figura 2.1 – Tela com o menu administrativo do PHP-Nuke (PHP-NUKE, 2003)

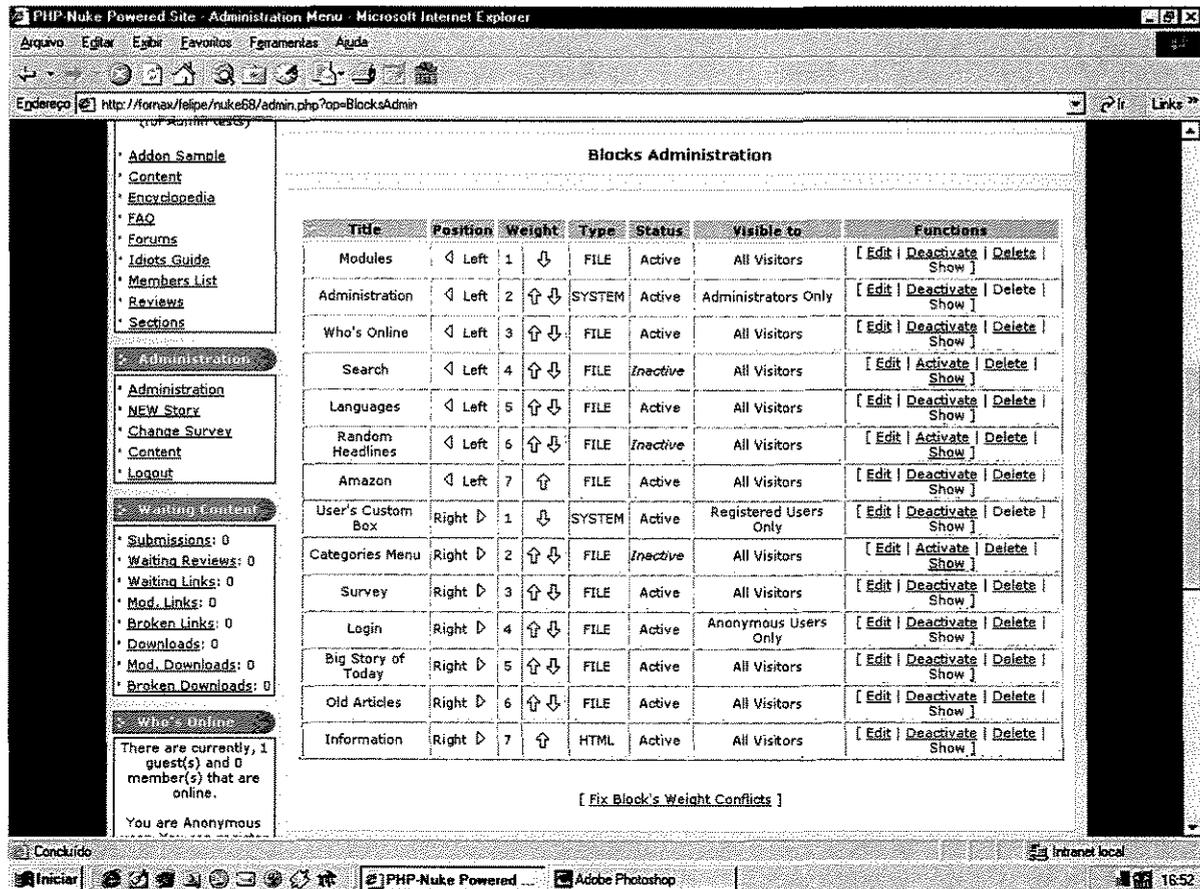


Figura 2.2 – Tela de administração dos Blocos do PHP-Nuke (PHP-NUKE, 2003).

2.5 O Plone

O Plone (Limi et al., 2003) foi desenvolvido sobre o Zope (Zope, 2003), uma ferramenta considerada bastante poderosa e complexa. O Zope foi desenvolvido na linguagem Python e utiliza conceitos de orientação a objetos; tudo que é criado dentro dele é tratado como objeto. Ele fornece subsídios para o desenvolvimento de aplicações web como um ambiente de desenvolvimento, um banco de dados nativo e uma linguagem própria, o DTML que pode ser usada para criação de *scripts* e também para a criação de

páginas HTML dinâmicas.

Assim como o Zope, o Plone é um software livre desenvolvido por uma comunidade de desenvolvedores (www.plone.org). Todo conteúdo criado dentro do Plone também é tratado como objeto. Ele é o CMS desenvolvido em Zope mais usado atualmente. Seu objetivo segundo seus autores é agregar valor em todos os níveis de uma organização (Limi et al., 2003). Almeja-se atingir esse objetivo por meio principalmente de um *workflow* para publicação das páginas.

Esse *workflow* é baseado no estado dos objetos e em regras para usuários que serão descritos a seguir. Existem quatro estados: visível, pendente, publicado e privado, e cinco tipos de regras: anônimos, membros, revisores, proprietários e gerentes.

Todo conteúdo do *site* (imagens, textos, arquivos, e outros) é considerado um objeto dentro do Plone. Os objetos podem assumir um dos estados citados.

No estado *visível* qualquer usuário pode visualizar o objeto por meio da função de busca ou digitando a URL do objeto diretamente no navegador. Entretanto, não existe um *link* para o objeto na árvore de navegação do *site*. Por padrão, quando um objeto é criado, assume o estado visível e só pode ser editado por seu proprietário e pelo gerente do *site*.

Objetos no estado “pendente” são objetos que foram submetidos para publicação e aguardam uma aprovação para serem publicados. Eles podem agora ser editados também pelos revisores que poderão aprová-lo ou rejeitá-lo.

No estado “publicado”, os objetos passam a ser visíveis também na árvore de navegação do *site* podendo ser editados apenas pelo gerente. O proprietário pode apenas reverter seu estado para visível.

O último estado é o “privado” onde o objeto passa a ser visível e editável apenas pelo seu proprietário e pelo gerente.

As regras definem o que diferentes categorias de usuários podem ver e fazer.

“Anônimos” são todos os usuários que estão acessando o *site* e não se identificaram. Usuários anônimos podem visualizar objetos nos estados publicado, pendente e visível, por meio da busca ou digitando a URL do objeto no navegador.

Usuários “cadastrados” que se identificaram no *site* têm privilégios que as regras de membros possibilitam. Eles são proprietários de uma pasta onde podem criar conteúdo no estado visível e submetê-lo para publicação.

O “revisor” pode editar um conteúdo submetido, aprová-lo ou rejeitá-lo.

As regras para “proprietário” são aplicadas aos conteúdos criados por um membro; ou seja, são as permissões que um membro possui em relação a um objeto criado por ele.

Os “gerentes” do *site* podem ver os conteúdos em todos os estados. Além das permissões de membro e de revisor o gerente também pode adicionar, editar, apagar e mover qualquer conteúdo. Gerentes também podem adicionar, editar e remover usuários e associar regras a eles.

A criação de conteúdo no *site* gerenciado requer que um usuário previamente cadastrado identifique-se. Para isso, ele deve clicar em no *link* acessar no canto superior direito da página principal do *site* e informar seu nome e senha.

Cada usuário com a permissão de membro no *site* possui uma pasta onde pode criar e armazenar conteúdo. Essa pasta fica localizada em outra pasta chamada “Membros” e para acessá-la é preciso clicar no *link* “Minha Pasta” localizado no canto superior direito do *site*

(Figura 2.3).

Para adicionar um conteúdo em sua pasta o usuário precisa ativar o modo “visão de conteúdo” clicando em um *link* ao lado esquerdo da tela. Feito isso, ele deve selecionar de uma lista qual é o tipo de conteúdo que deseja incluir e clicar no botão “*adicionar novo item*” (Figura 2.3). Nesse ponto é aberta a página de edição (Figura 2.4) do conteúdo de acordo com o tipo selecionado e dar-se-á início à criação efetiva do conteúdo pelo usuário. A página de edição contém quatro abas selecionáveis. A primeira exibe uma pré-visualização do conteúdo que está sendo editado, a segunda (Figura 2.4) exibe o formulário de edição do conteúdo, a terceira exibe as propriedades (ou metadados) associadas àquele conteúdo como idioma, palavras chaves, formato, *copyright* e outras dependendo do tipo de conteúdo. A quarta (Figura 2.5) permite mudar o estado do conteúdo dando início ao processo de publicação onde ele será revisado por um editor e finalmente publicado.

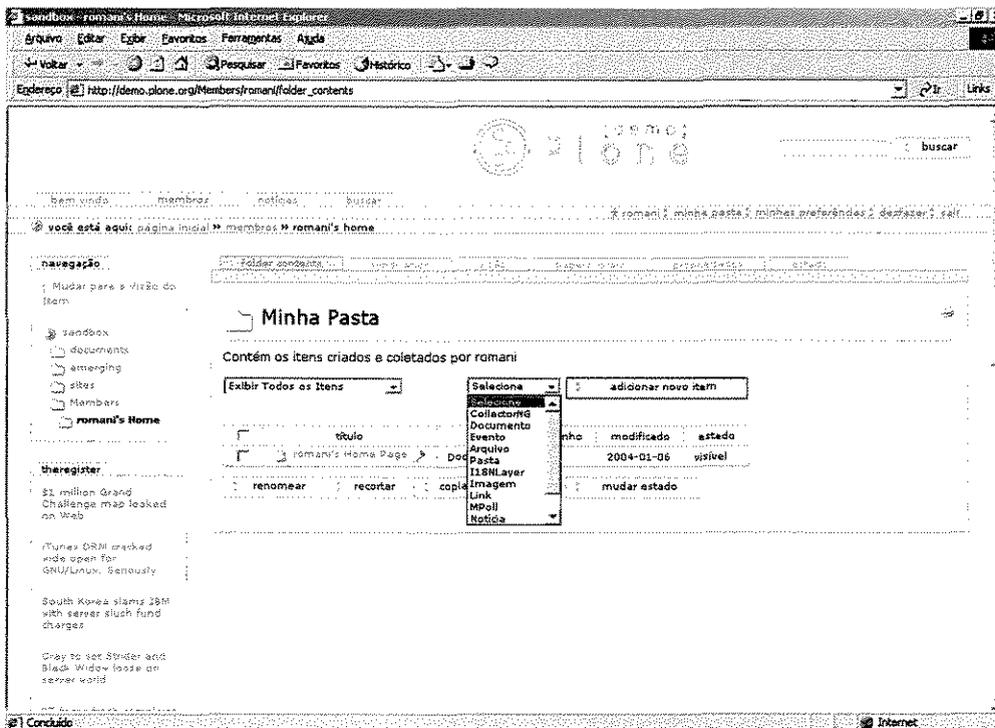


Figura 2.3 – Tela com a pasta de um usuário membro do Plone (PLONE, 2003).

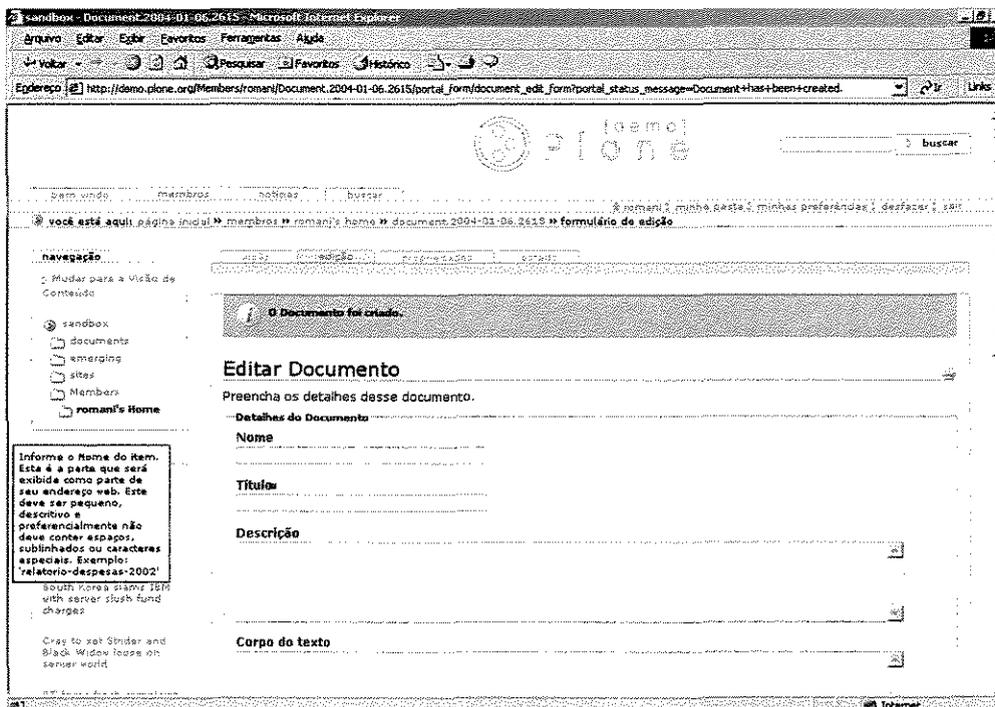


Figura 2.4 – Tela de edição de um conteúdo recém criado no Plone (PLONE, 2003).

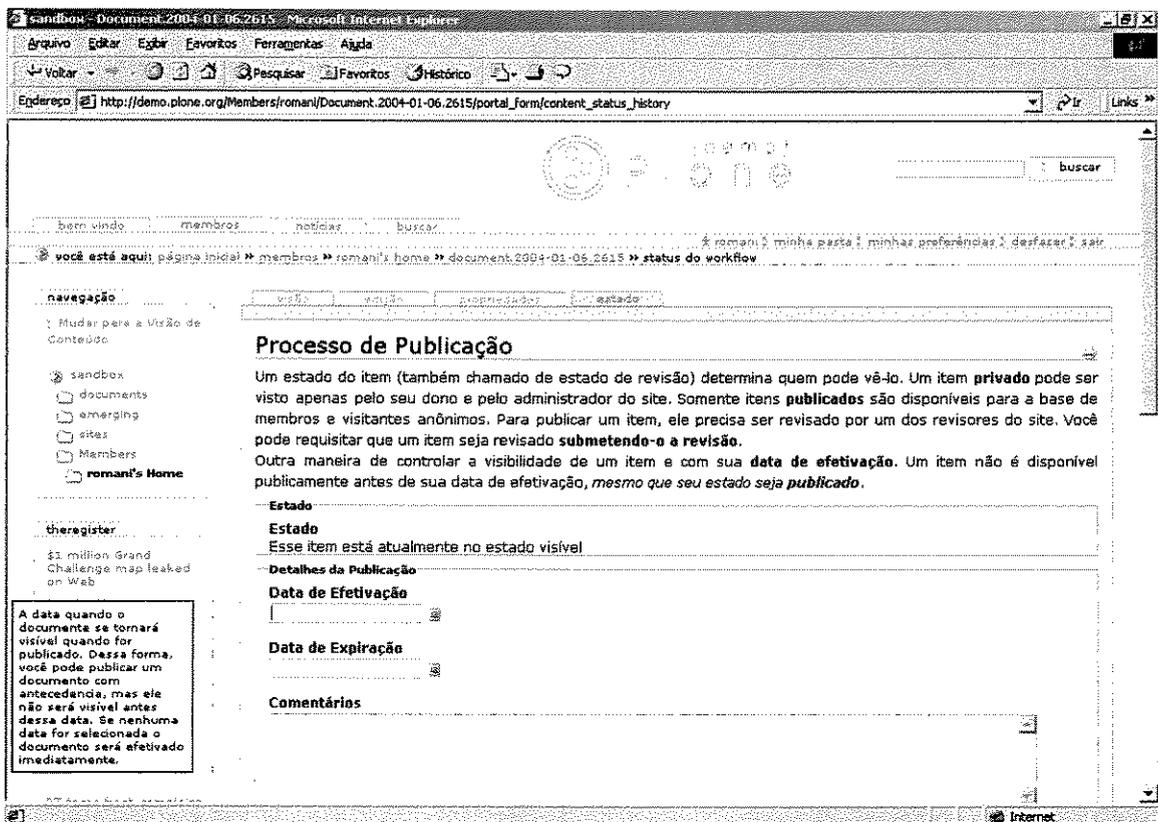


Figura 2.5 – Tela que permite alterar o estado do conteúdo no Plone (PLONE, 2003).

2.6 Porque desenvolver outro CMS?

Além do PHP-Nuke e do Plone foram analisados os seguintes CMS, com o propósito de identificar algum que atendesse as necessidades já mencionadas no capítulo 1: OpenCMS e Cocoon desenvolvidos na linguagem de programação Java, PostNuke e Xoops desenvolvidos em PHP, Squishdot desenvolvido em Zope, Slach e WebMake desenvolvidos na linguagem Perl.

A maioria deles dispensou uma análise mais detalhada por serem muito específicos, ou seja, feitos para um determinado tipo de *site* como por exemplo sites de notícias ou por serem demasiadamente complexos para o tipo de usuário que se pretendia atingir.

Em todos os CMS analisados as alterações de *layout* do *site* só podem ser feitas por um usuário administrador ou gerente que precisa pelo menos conhecer HTML ou alguma linguagem de programação, dependendo da complexidade da alteração. Com isso é necessário que o administrador do *site* seja um técnico em informática.

Outra limitação é que, para cada *site* gerenciado é preciso fazer uma nova instalação do CMS o que no caso de se hospedar vários *sites*, aumenta consideravelmente o trabalho de suporte pois uma atualização de segurança, por exemplo, deverá ser executada para cada um dos *sites* individualmente.

Para exemplificar, suponhamos que uma unidade acadêmica de uma universidade queira usar um CMS para gerenciar os *sites* de todos os professores ou grupos de pesquisa. Cada *site* precisará de uma pessoa com conhecimentos técnicos para administrá-lo.

O principal problema identificado no PHP-Nuke foi a complexidade de sua interface em função do tipo de usuário que se pretende atingir nesse trabalho.

Com relação ao Plone sua interface é mais intuitiva porém o processo de criação e publicação de conteúdos está fortemente vinculada a seu *workflow* criando assim uma dificuldade desnecessária para o tipo de site e de usuário alvo desse estudo.

2.7 Considerações finais

Nesse capítulo foi descrito o funcionamento básico dos CMSs que surgiram em substituição ao uso de páginas HTML estáticas, facilitando a tarefa de gerenciamento de *sites*.

Foi dado um maior destaque a dois gerenciadores conhecidos desenvolvidos na linguagem de *script* PHP e na plataforma Zope de desenvolvimento de aplicações web.

No próximo capítulo será descrito em detalhe o GECON uma proposta de CMS mais simples, voltada a pessoas sem conhecimentos técnicos em tecnologia para web e que oferece um nível de manutenção constante independente do número de *sites* hospedados.

Capítulo 3

GECON – GEreenciador de CONteúdos: Design e Desenvolvimento

Introdução

O objetivo geral do GECON é simplificar o processo de construção e gerenciamento de *websites* de forma colaborativa. Ele possibilita que, mesmo pessoas sem conhecimento técnico em tecnologia da informação, tenham a possibilidade de administrar seus *websites* sem o auxílio de desenvolvedores ou técnicos nessa tarefa. Boa usabilidade e simplicidade foram as metas estabelecidas.

Ele foi desenvolvido pela equipe de EAD da Unicamp tendo como resultado prático imediato facilitar a manutenção do *site* de EAD Unicamp e atender a uma demanda crescente de hospedagem de *sites* ligados a EAD.

3.1 Aspecto Funcional

O GECON está organizado por sete elementos distribuídos em três regiões da tela: cabeçalho, corpo e rodapé, como pode ser visto na Figura 3.1. Na região do cabeçalho estão localizados os elementos: Logo que contém a imagem de logotipo do *site*, Banner que pode conter um texto ou uma imagem, e Menu Horizontal onde fica a navegação principal do *site*. A segunda região, é denominada corpo, possui os elementos: Menu Vertical contendo a navegação secundária, Área Principal onde é apresentado o conteúdo das páginas e Parcerias que pode conter uma lista de imagens ou texto com *links*. para outros sites. A terceira é a região do rodapé, que possui apenas um elemento que também leva o nome de Rodapé onde são colocadas informações que serão visualizadas em todas as páginas do *site*.

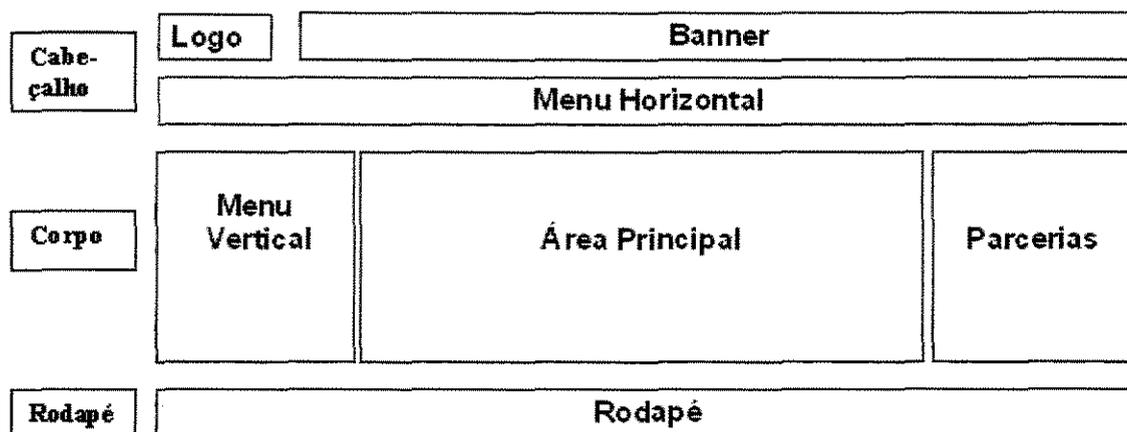


Figura 3.1 – Elementos do GECON divididos por regiões

A localização dos elementos na tela foi definida com base em um estudo estatístico feito por Nielsen (NIELSEN, 2001) mostrando em qual região da tela cada elemento é encontrado, predominantemente. Com isso o visitante do *site* pode fazer uso de sua experiência adquirida em outros *sites* facilitando assim a navegação.

O *layout* é fluido, ou seja, ele se adapta automaticamente ao tamanho da janela do navegador do usuário. Ao contrario de layouts congelados que permanecem idênticos em janelas grandes e pequenas, o fluido evita desperdício de espaço em janelas maiores e barra de rolagem horizontal em janelas menores do que o tamanho em que o *layout* foi congelado.

Qualquer subconjunto dos elementos pode ser utilizado para montar o *site*, com isso é possível atender diferentes necessidades do usuário em relação ao *layout*. A disposição é fixa, ou seja, não é possível mover os elementos de uma área para outra. Entretanto pode haver alteração em suas dimensões.

Essa característica de seleção de alguns elementos em conjunto com a possibilidade de alteração de seus tamanhos provê flexibilidade e permite criar *sites* visualmente diferentes.

Administração

Toda a administração do *site* é feita via navegador; com isso, o administrador do *site* independe de outras pessoas para fazer *uploads* ou acessar áreas restritas. Além disso, dispensa a necessidade de conhecimentos de outros software para fazer o gerenciamento.

O acesso ao *site* como administrador é feito acrescentando-se */login* no final do seu endereço. É solicitada então a identificação e senha do administrador e o *site* passa a ser visualizado com a interface administrativa que consiste de um menu com duas opções: *Geral* e *Ajuda* e um *ícone* de edição para cada elemento conforme ilustrado na Figura 3.2.

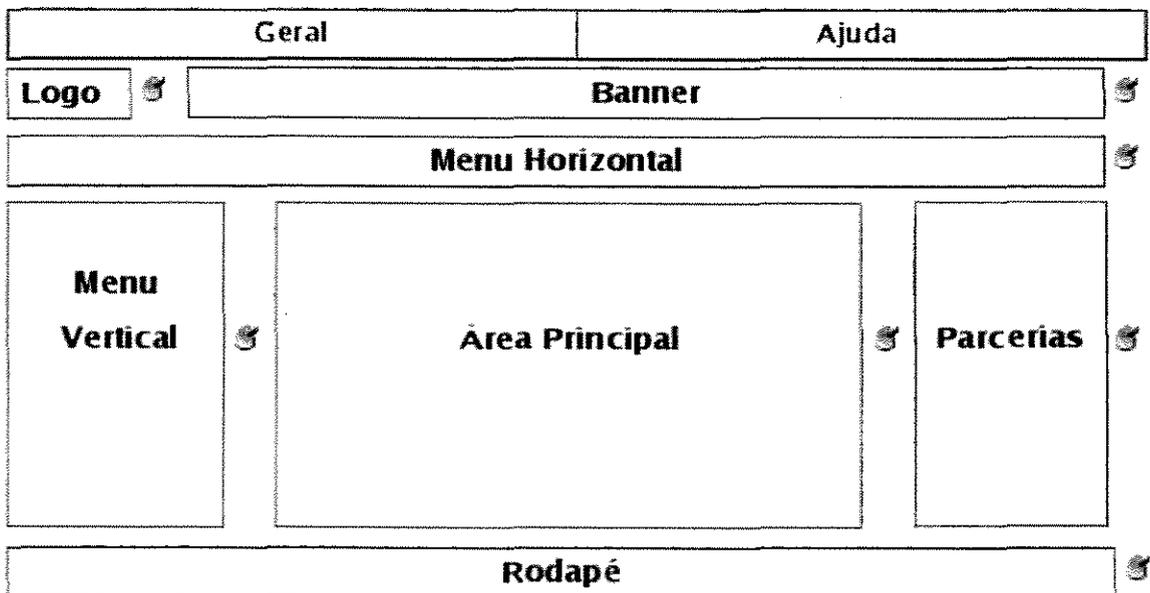


Figura 3.2 – Interface administrativa do GECON

Ao se clicar em um *ícone*, é aberta uma nova janela que possibilita a edição do elemento associado àquele ícone. Essa janela de administração é específica para cada elemento. No entanto, ela apresenta um menu de navegação com acesso à janela administrativa de todos os elementos, o que permite ao usuário navegar entre elas sem ter que voltar à janela do *site*. A Figura 3.3 ilustra a janela administrativa da opção *Geral*.

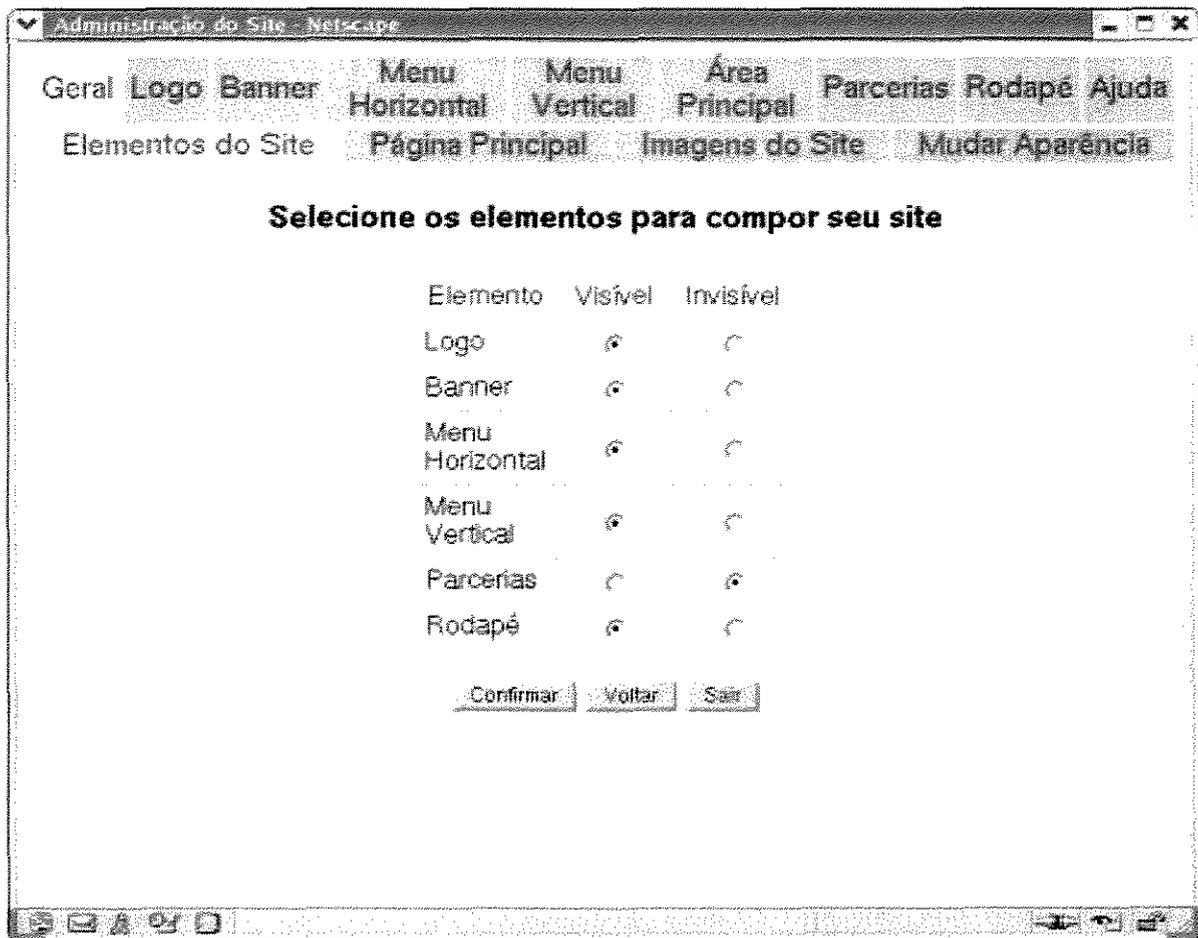


Figura 3.3 -Janela administrativa da opção Geral

A opção *Geral* da interface administrativa é usada para as configurações que se aplicam ao *site* como um todo como por exemplo definir qual subconjunto de elementos será usado no *site*. A opção *Ajuda* traz um texto explicando como utilizar o GECON. A seguir descreveremos em detalhe cada uma das opções da interface administrativa.

A opção *Geral*

Ao selecionar a opção *Geral*, abre-se a janela administrativa ilustrada pela Figura 3.3 contendo 4 subpastas. A primeira delas, designada *Elementos do Site* permite selecionar quais elementos serão utilizados para compor o *site*. Para isso basta defini-los como visíveis.

A segunda pasta, designada *Página Principal*, permite configurar a página inicial do *site*. Nela é apresentada uma lista com todas as páginas criadas até o momento para que o usuário selecione qual será a página inicial. Além disso, a lista oferece uma opção especial designada “página padrão”. Essa opção torna a página principal do *site* uma página gerada automaticamente contendo as novidades do *site*.

A pasta *Imagens do Site* permite gerenciar todas as imagens do *site*. É possível incluir e excluir imagens que posteriormente serão referenciadas nas páginas HTML do *site*.

A última pasta da opção *Geral* é a pasta *Mudar Aparência* que possui um formulário contendo vários itens do *site* que alteram sua aparência geral. São eles: nome do *site*, imagem de fundo, cor de fundo, tipo de fonte, tamanho da fonte, cor do link, cor do link já visitado, cor do link quando apontado pelo mouse e cor do link quando ativado.

Essas opções se aplicarão a todos os elementos do *site* a menos que sejam redefinidas na pasta *Mudar Aparência* do elemento.

Elemento *Logo*

O elemento *Logo* foi criado para armazenar o logotipo do *site*. Sua janela administrativa apresenta apenas uma subpasta que apresenta um formulário por meio do qual é possível importar uma imagem, alterar a cor de fundo e alterar o tamanho do elemento *Logo*.

À imagem do *Logo* está associado um *link* para a página principal do *site*.

Elemento *Banner*

No *Banner*, além da possibilidade de incluir uma imagem, sua janela administrativa (Figura 3.4) permite opcionalmente incluir um texto. Essa funcionalidade possibilita que mesmo aquele usuário que não possua, não saiba ou não queira trabalhar com imagens possa fazer uso deste elemento.

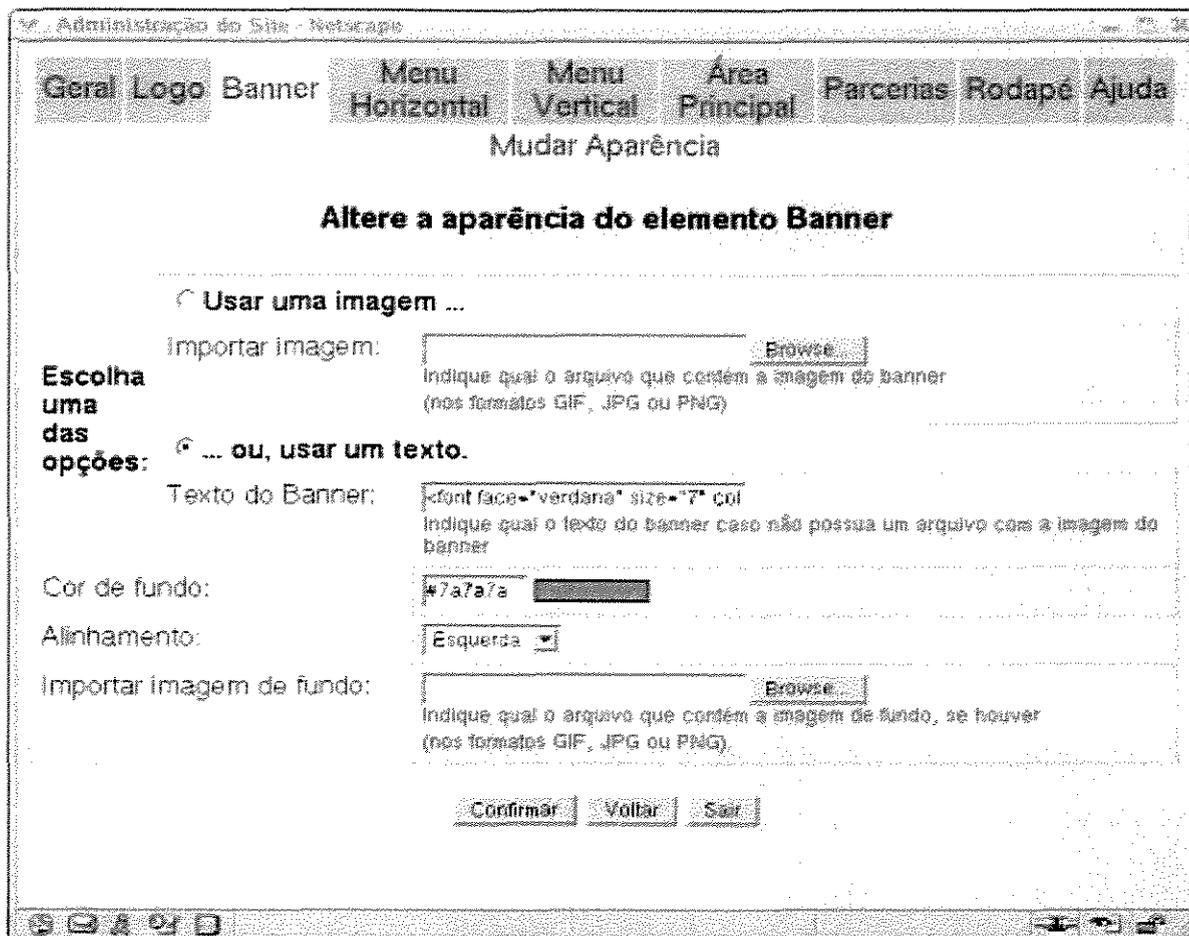


Figura 3.4 – Janela administrativa do elemento Banner

No texto do *Banner* é permitido o uso de tags HTML; com isso o usuário pode aumentar o tamanho, cor e tipo de fonte.

A cor de fundo do *Banner* pode ser alterada diretamente no formulário. Usuários avançados podem digitar o código hexadecimal da cor no padrão RGB, mas também pode-se fazer uso do mapa de cores que abre uma tabela para selecionar a cor desejada.

Elemento Menu Horizontal

O *Menu Horizontal* é o terceiro elemento do cabeçalho. Sua função é permitir ao usuário criar um menu principal que, assim como os outros dois elementos do cabeçalho, será apresentado em todas as páginas do *site*. Sua janela de administração (Figura 3.5) está dividida em duas “pastas”: a primeira é a que define os itens do menu e a segunda serve para definir sua aparência.

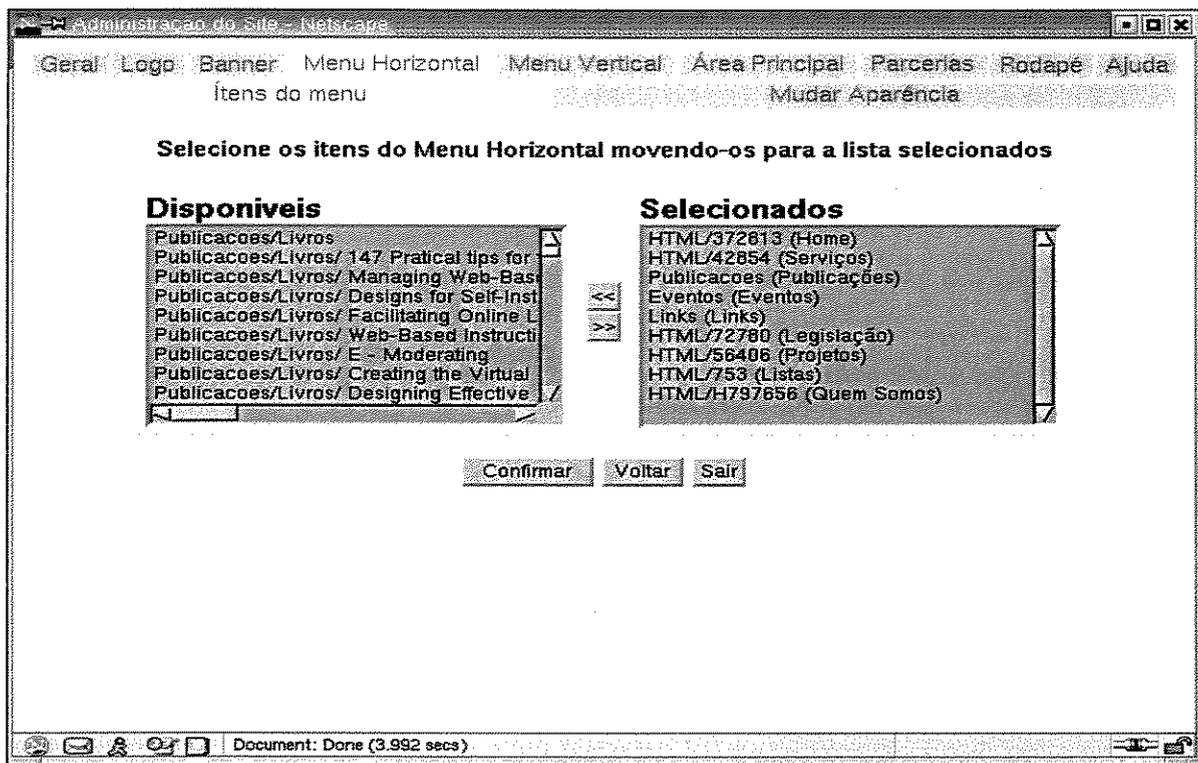


Figura 3.5 – Janela administrativa do Menu Horizontal

A pasta “itens do menu” apresenta duas listas: a *Disponíveis*, contendo todas as páginas e pastas criadas no *site* até o momento e a *Selecionados* com as páginas e/ou pastas que foram selecionadas para compor o menu.

Para incluir um item no menu, basta selecioná-lo da lista *Disponíveis* e movê-lo para a lista *Selecionados* clicando no botão “>>”. Do mesmo modo, para excluir um item do menu, basta selecioná-lo na lista *Selecionados* e movê-lo para a lista *Disponíveis* clicando agora no botão “<<.”

A segunda pasta da janela administrativa do menu horizontal é chamada “Mudar Aparência” Quase todos os elementos possuem uma pasta mudar aparência; como o próprio nome sugere, ela serve para alterar a aparência do elemento. No menu horizontal é possível alterar: o tipo de fonte, o tamanho de fonte, colocar negrito, a cor do link, a cor do link já visitado, a cor do link quando apontado pelo mouse, a cor do link quando ativado, a cor de fundo e a imagem de fundo.

No menu cada item é um link e não é possível definir uma imagem para cada item do menu.

Elemento *Menu Vertical*

O Menu Vertical tem a função de menu secundário; por exemplo, suponha que se queira colocar um item publicações no menu principal e que as publicações tenham sido organizadas em várias páginas separadas cada uma com um tipo de publicação diferente como artigos em uma página, teses em outra, e relatórios em uma terceira página. É natural que se queira criar um menu secundário com esses 3 tipos de publicações. É exatamente esta a função do menu vertical e seu funcionamento é automático.

O menu vertical funciona da seguinte maneira: sempre que for criado um item do tipo pasta no menu principal, o menu secundário exibirá o conteúdo da pasta como sendo seus itens. Tomando o exemplo das publicações, se for criada uma pasta Publicações, contendo

3 arquivos denominados Artigos, Teses e Relatórios, ao selecionar a pasta Publicações como item do menu principal, automaticamente aparecerá no menu vertical 3 itens: Artigos, Teses e Relatórios.

Pelo fato do menu vertical ser criado automaticamente, sua janela administrativa (Figura 3.6) possui apenas a pasta mudar aparência que permite alterar o tipo de fonte, o tamanho da fonte, colocar negrito, a cor do link, a cor do link já visitado, a cor do link quando apontado pelo mouse, a cor do link quando ativado e a cor de fundo.

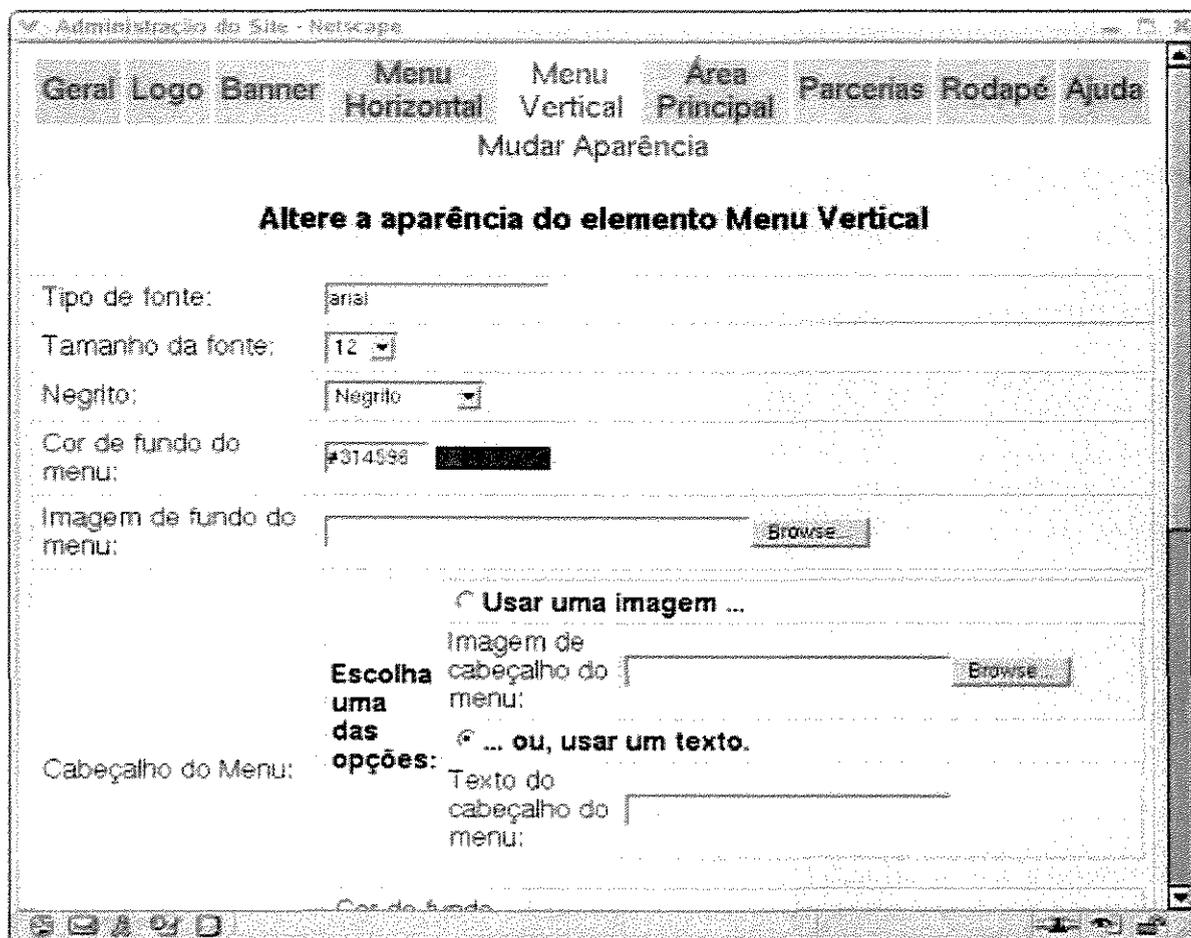


Figura 3.6 – Janela administrativa do Menu Vertical

Elemento *Área Principal*

Esse é o elemento central do *site*: área reservada para apresentar o conteúdo do *site*. Sua janela administrativa está subdividida em quatro assistentes que auxiliam o usuário de acordo com o tipo de conteúdo a ser trabalhado que na primeira versão do sistema podem ser: publicações, eventos, links ou páginas HTML sem uma pré-formatação específica. Outros assistentes podem ser implementados e facilmente incorporados ao GECON em versões futuras, mas todos os assistentes apresentam o mesmo comportamento, descrito a seguir.

Os conteúdos ficam organizados de forma hierárquica em uma estrutura de pastas e documentos, similar à de um gerenciador de arquivos; porém, no assistente, a pasta leva o nome de categoria e o documento corresponde ao conteúdo, depende do tipo de assistente (HTML no assistente de HTML, Evento no assistente de Eventos, Publicação no assistente de Publicações e Links no assistente de Links).

Tomemos como exemplo o assistente de HTML. No primeiro acesso ao assistente existe apenas a categoria raiz HTML. Ao selecionar a categoria HTML abre-se um menu com as opções para o usuário criar uma nova categoria (pasta) ou um novo conteúdo do tipo HTML. Selecionada a opção desejada, será apresentado um formulário para o preenchimento das informações necessárias para se criar a categoria ou o conteúdo.

Todas as categorias e conteúdos criados podem ser excluídos, movidos ou editados por meio das opções de um menu que é apresentado ao usuário.

As categorias podem ser abertas ou fechadas clicando-se nos ícones “+” ou “-” respectivamente (Figura 3.7). Quando aberta, a categoria exibe seu conteúdo que pode ser composto de outras categorias ou conteúdos.

Dessa forma o usuário pode criar seu conteúdo de acordo com suas necessidades de organização. Na Figura 3.7 é apresentada a tela do assistente de HTML com diversas categorias e documentos criados.

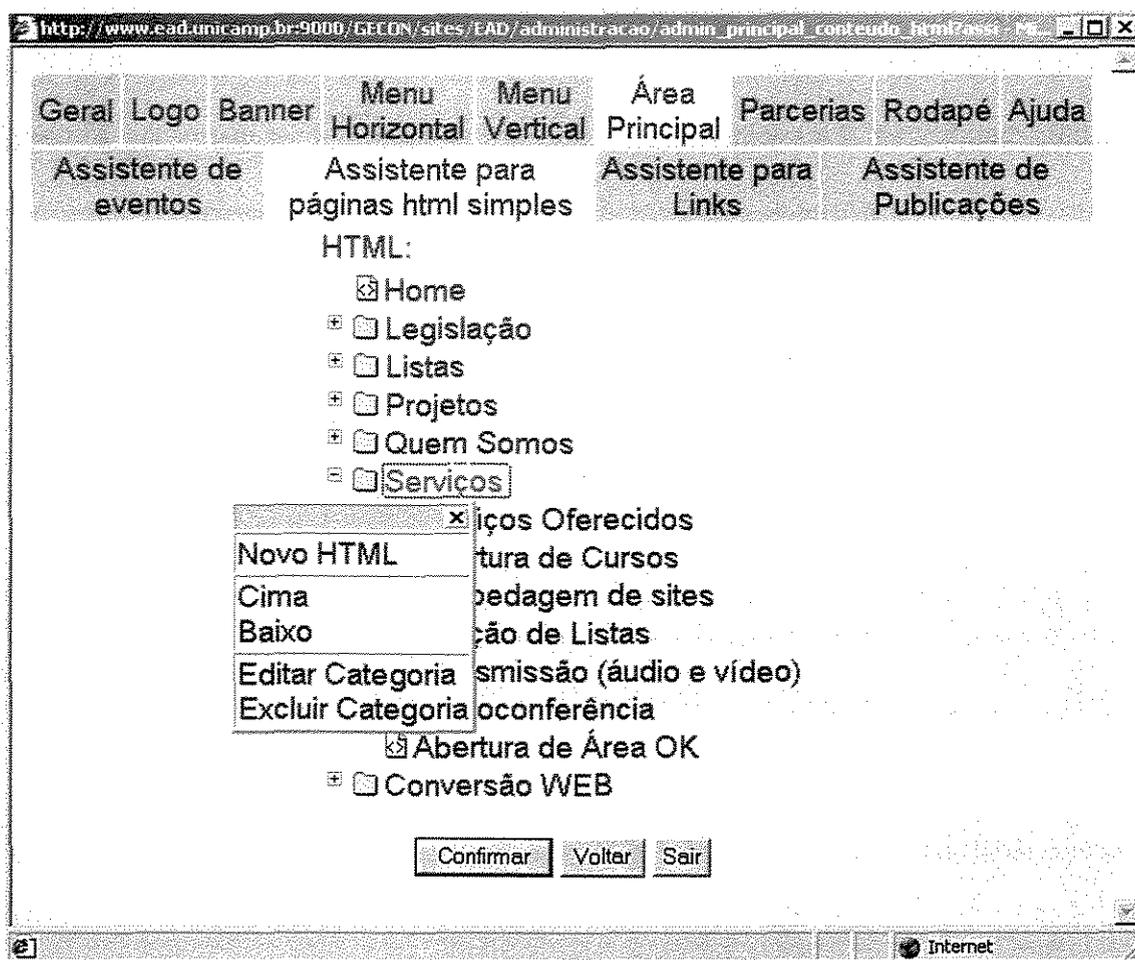


Figura 3.7 – Assistente de HTML

As opções *Cima* e *Baixo* que aparecem no menu da Figura 3.7 permitem mover as categorias e conteúdos dentro da hierarquia criada.

Nessa primeira versão do GECON, são oferecidos quatro tipos de assistentes: HTML, Eventos, Links e Publicações. Novos assistentes podem ser implementados para diferentes

tipos de conteúdo, aumentando assim as funcionalidades do sistema, adaptando-o a novas necessidades dos usuários.

Elemento *Parcerias*

Esse elemento fica localizado na parte direita da tela; nele é possível colocar imagens com links para outros *sites* ou para páginas do próprio *site* que se queira dar um grande destaque. Suponha que esteja sendo criado um *site* de um projeto financiado por vários órgãos de fomento; poder-se-ia colocar os logotipos dos *sites* financiadores apontados para *sites* daquelas instituições.

Sua janela administrativa possui 3 pastas, *Incluir parceiros* é a pasta que contém um formulário para importar uma imagem para o *site*, atribuir um nome que aparecerá abaixo da imagem, caso isso seja necessário, e uma URL que será o *link* da imagem; a pasta *excluir parceiros* apresenta a lista dos parceiros já importados para o *site* para que se selecione aquele que se deseja excluir; a pasta *mudar aparência* permite incluir um título que aparecerá no topo do elemento, além de alterar as cores de fundo e da borda do elemento *parcerias*.

Elemento *Rodapé*

O último elemento é o *Rodapé*, que possui apenas uma pasta para se incluir o texto do rodapé e alterar a cor de fundo.

O texto do rodapé aceita marcações HTML que permitem ao usuário formatá-lo da maneira que desejar.

Assim como os elementos *Logo*, *Banner* e *Menu Horizontal*, o *Rodapé* estará presente em todas as páginas do *site*.

3.2 Aspecto Arquitetônico

O GECON foi desenvolvido em Zope Z Object Publishing Environment (Zope, 2003). Zope é um software livre (código aberto), criado pela Digital Creations (atual Zope Corporation). Ele permite gerenciar e desenvolver *sites* e aplicações web com conteúdo dinâmico, ou seja, as páginas HTML não são estáticas, mas criadas a partir de dados armazenados em um banco de dados no momento em que o servidor web recebe uma solicitação do navegador para exibi-las.

O Zope possui funcionalidades similares às das linguagens PHP, ASP ou Perl, mas tem como principal vantagem a facilidade de atualização do conteúdo devido a sua interface similar a de um gerenciador de arquivos. Essa interface é acessada através de um navegador e faz uso de *frames*, *logins* e *senhas*.

Zope é baseado nos conceitos de Orientação a Objetos (OO), o que requer um mínimo de familiaridade com os conceitos de OO para que se possa explorá-lo satisfatoriamente. Isso dificulta seu uso por profissionais com poucos conhecimentos técnicos em informática.

Ele é multiplataforma ou seja é possível executá-lo em diversos sistemas operacionais. No *site* do Zope (<http://www.zope.org>) estão disponíveis versões compiladas para Linux, Solaris e Windows. Para outros sistemas operacionais é preciso “baixar” o código fonte e compilá-lo. Ele possui uma linguagem própria de programação de alto nível chamada DTML(Document Template Markup Language) mas como foi escrito na

linguagem Python, permite também a criação de *scripts* nessa linguagem para resolver problemas mais complexos.

Todos os objetos criados no Zope são armazenados em um banco de dados de onde são recuperados para compor as páginas dinamicamente no navegador. Embora o Zope possua seu próprio banco de dados orientado a objetos, é possível utilizá-lo com Oracle, MySQL, SQL Server e outros.

Os fatores fundamentais para a escolha do Zope como base para o desenvolvimento do GECON foram o fato de ser software livre, multiplataforma, orientado a objetos, com capacidade de atualização de conteúdo via navegador, armazenamento dos objetos em banco de dados e ainda poder escolher qual banco usar.

Podemos visualizar o GECON como uma camada entre o Zope e o administrador do *site* (Figura 3.8). Essa camada tem por objetivo viabilizar o uso de uma ferramenta extremamente poderosa, porém muito complexa, para usuários que não possuem o conhecimento técnico necessário.

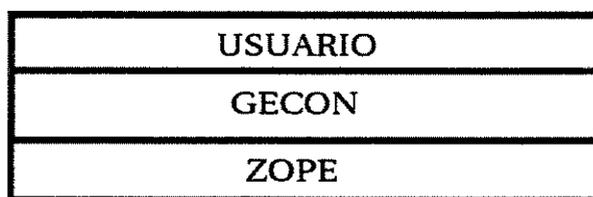


Figura 3.8 – GECON entre o administrador do *site* e o Zope

Estrutura

O GECON está hierarquicamente representado como uma estrutura de diretórios onde aqueles situados em níveis mais baixos na hierarquia herdam os métodos dos diretórios de níveis mais altos, conforme ilustra a figura 3.9

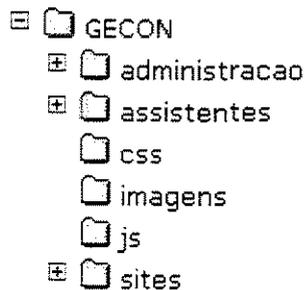


Figura 3.9 – Estrutura hierárquica do GECON

Cada diretório (ou pasta) do GECON pode conter propriedades, métodos e outros subdiretórios (ou subpastas).

A pasta raiz denominada GECON possui os seguintes métodos, que podem ser usados em qualquer subpasta:

`cabeçalho_html` – monta a região denominada cabeçalho das páginas dos sites hospedados; o cabeçalho contém os elementos *Logo*, *Banner* e *Menu Horizontal*.

`corpo_html` – monta a região denominada corpo das páginas dos sites hospedados; o corpo contém os elementos *Menu Vertical*, *Área Principal* e *Parcerias*.

`rodapé_html` - monta a região denominada rodapé das páginas dos sites hospedados; essa região contém apenas o elemento rodapé.

`menuh_html` – monta o elemento *Menu Horizontal*.

menuv_html - monta o elemento *Menu Vertical*.

parcerias_html - monta o elemento *Parcerias*.

principal_html - monta o elemento *Área Principal*.

login - apresenta a tela para identificação do usuário administrador do *site*.

mapa_site – Gera uma página com o mapa do *site*.

Além de métodos, o diretório GECON possui as seguintes subpastas:

administração - armazena todos os métodos e páginas usadas para a administração dos *sites*,

assistentes - contém os assistentes para ingresso dos diferentes tipos de conteúdos já mencionados,

css - contém uma folha de estilo geral,

imagens: - contém apenas imagens usadas pelo GECON. As imagens dos *sites* criados ficam armazenadas no subdiretório imagens correspondente a cada *site*,

js - contém os *scripts* em javascript usados no GECON,

sites - neste diretório ficam hospedados os *sites*. Cada *site* é um subdiretório dentro do diretório *sites*.

A seguir é detalhada cada uma dessas pastas do nível 1 da hierarquia:

Pasta administração

Na pasta administração ficam armazenados os métodos que geram as janelas administrativas de todos os elementos (logo, banner, menu horizontal, etc.)

Uma janela administrativa é montada da seguinte maneira:

Existem dois métodos `cabeçalho_admin_html` e `rodape_admin_html` que são comuns a todas as janelas administrativas dos sete elementos. No cabeçalho é montado o menu para navegação entre as janelas administrativas e no rodapé os botões para cancelar ou confirmar as ações efetuadas na janela. A parte central é montada por métodos específicos que levam o nome dos elementos.

Além dos métodos que geram as janelas administrativas, cada elemento possui um método que recebe as informações da janela quando é selecionado o botão confirma.

Existem duas subpastas na pasta administração. A de *métodos_auxiliares* contém métodos invocados pelos métodos da administração e a subpasta *css* contém um arquivo de definições de folha de estilo para as janelas administrativas.

Pasta assistentes

Contém os módulos que auxiliam o administrador do *site* a incluir e organizar conteúdos de tipos específicos, como por exemplo uma página de links sugeridos ou uma página de eventos (congressos, encontros, *workshops*, etc.).

A primeira versão do GECON oferece cinco assistentes mas novos assistentes podem ser criados e anexados ao GECON expandindo-se e aperfeiçoando-se o sistema.

Cada assistente tem sua subpasta própria e possui a seguinte estrutura básica: dois métodos para criar o conteúdo e um método para apresentar o conteúdo do *site*.

O método que apresenta o conteúdo do *site* deve ter o nome “`exibir_conteudo`”.

Os métodos que criam o conteúdo devem ter o nome “`novo_conteudo_XYZ`” onde XYZ é o tipo do conteúdo gerenciado pelo assistente e “`novo_conteudo_target`”. O

primeiro gera a janela para o administrador do *site* criar o conteúdo e o segundo recebe os dados informados na janela e efetivamente cria o conteúdo.

Todo o conteúdo criado via um assistente fica armazenado na subpasta conteúdos do *site* (dentro da pasta *sites* do GECON)

Pasta CSS

Esta pasta contém uma folha de estilo geral. Cada *site* também possui sua própria folha de estilo, o que permite que cada um tenha sua identidade visual própria. Porém, se o GECON não encontrar a definição de uma classe css na folha de estilo do *site*, ele buscará essa definição na folha de estilo geral, localizada nessa pasta.

Pasta imagens

Contém apenas as imagens usadas pelo GECON como o ícone que aparece ao lado de cada elemento na visão do administrador. As imagens dos *sites* ficam armazenadas no subdiretório *imagens* dentro de cada *site*.

Pasta js

É uma pasta auxiliar que contém os *scripts* em javascript usados no GECON. A palheta de cores apresentada para facilitar o preenchimento dos campos com o código das cores por exemplo é feita em javascript, existem também *scripts* de verificação dos dados fornecidos nos diversos formulários.

Pasta sites

Esta pasta contém os *sites* gerenciados pelo GECON. Cada *site* fica hospedado em uma subpasta que leva o nome do *site*. Dentro da pasta do *site* existe um método *index_html* que monta dinamicamente não só a página inicial, mas todas as páginas do *site*. O *index_html* usa os métodos definidos na pasta raiz GECON para montar as páginas do *site*. Esses métodos quando executados a partir de um *site* buscam as informações no diretório do *site* que os invocou para exibi-las. Sendo assim, ao invocar o método “cabeçalho_html” a partir do *site* EAD o método exibirá o cabeçalho do *site* EAD. Dessa forma não é preciso replicar métodos dentro dos *sites*.

Os dados do *site* ficam armazenados exclusivamente dentro da sua pasta. Eles podem ser propriedades da pasta como por exemplo os itens do menu horizontal, ou podem ser arquivos como por exemplo uma página criada pelo administrador ou ainda imagens como o logo do *site*. Para organizar essas informações cada *site* possui a seguinte estrutura: uma pasta de usuários (*acl_users*), uma pasta de conteúdos, uma pasta *css* e uma pasta *imagens*, além do método “*index_html*” já mencionado, que fica na pasta raiz do *site*, como ilustrado a seguir:

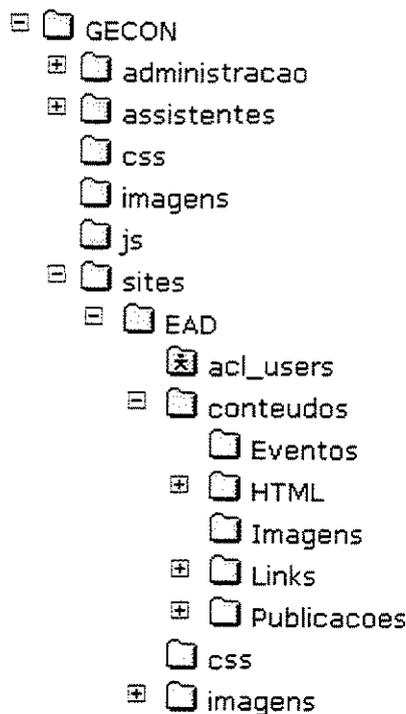


Figura 3.10 – Estrutura hierárquica expandida

A pasta *acl_users* contém a lista de usuários do *site*. Podem existir três tipos de usuários. O administrador (podendo haver mais de um), os visitantes autorizados e os visitantes padrão. A lista de usuários armazena apenas o administrador e os visitantes autorizados; qualquer visitante que não esteja na lista é considerado visitante padrão. Este último navega por todas as partes do *site* que não possuem restrição de acesso.

A pasta *conteúdos* armazena todo o conteúdo criado via assistentes. Dentro dele existe uma subpasta para organizar separadamente o conteúdo criado via cada assistente.

A pasta *css* possui a folha de estilo do *site* que é criada e alterada exclusivamente através das janelas administrativas de cada um dos elementos denominadas *alterar aparência*.

A pasta imagens armazena as imagens dos elementos Logo, Banner e Parcerias, inseridas através da janela administrativa de cada um desses elementos.

3.3 Considerações Finais

Como o GECON foi desenvolvido como uma camada sobre o Zope, todo o suporte à ferramenta é feito por meio da interface administrativa do Zope que assim como no GECON também é feita remotamente via navegador, portanto, tarefas como: corrigir erros, implementar melhorias e criar novos sites precisam de uma pessoa com conhecimentos técnicos suficientes para operar o Zope.

Todas as tarefas relacionadas a criação de conteúdo e design do *site* são realizadas pelo(s) usuário(s) cadastrado(s) como administrador(es) do *site* por meio do preenchimento de formulários, evitando assim a necessidade de conhecimentos técnicos por parte desse usuário. O acesso aos formulários é rápido e direto e todos os usuários administradores de *sites* possuem permissão para alterar qualquer aspecto dentro do *site* ao qual administra.

Nos outros CMSs analisados, tarefas relacionadas ao design dos *sites* são em geral complexas. Até mesmo a criação de conteúdo requer uma série de etapas como no caso do Plone - descritas no capítulo 2. Certas tarefas requerem reprogramação de partes do sistema, como no caso do PHP-Nuke, o que requer a intervenção de especialistas em informática.

Outra vantagem do GECON em relação aos outros dois sistemas analisados é a sua facilidade de manutenção. Montou-se a estrutura do GECON, explorando-se o recurso de herança do Zope de modo que um mesmo método fosse usado por todos os *sites* hospedados. Dessa forma, ao ser feita uma correção ou melhoria nesses métodos todos os

sites hospedados são beneficiados simultaneamente sem que o trabalho precise ser refeito para cada *site*.

No capítulo 4 serão apresentados os testes e análises da interface feitas durante o desenvolvimento do GECON que permitiram buscar um alto grau de facilidade de uso.

Capítulo 4

Análise da Interface do GECON: inspeção e teste

Introdução

A Avaliação é parte integrante do processo de design de software e deve ter lugar ao longo de seu ciclo de vida. Portanto, ela não deve ser pensada como uma fase única no processo de desenvolvimento do software, nem como uma atividade a ser realizada ao final do processo, se o cronograma permitir (Rocha e Baranauskas 2000). Fazendo um paralelo com os novos conceitos de qualidade em manufatura, a avaliação deve ocorrer durante o processo de design, com resultados gerando melhorias contínuas e alimentando modificações no produto final - o software.

Embora o usuário final em seu ambiente de trabalho seja a razão primeira de se pensar em qualidade da interface, em geral, não é possível realizar testes experimentais extensivos contínuos durante o design. Neste capítulo são descritos os métodos de inspeção de usabilidade utilizados na avaliação da interface do GECON e discutidos os resultados dessas análises. Foi utilizado o método avaliação heurística para tal análise.

O objetivo da aplicação de métodos de inspeção de usabilidade é identificar problemas na interface do sistema para corrigi-los em iterações subseqüentes no processo de design.

Também foi realizado um teste de uso do sistema que gerou uma série de recomendações complementares aos resultados da inspeção.

4.1 Avaliação Heurística

A seguir é feita uma descrição do método, do contexto no qual ele foi aplicado e apresentados os resultados obtidos com a avaliação.

4.1.1 Descrição do método

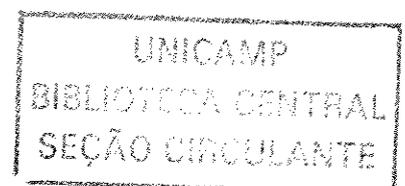
A avaliação heurística, também chamada “*discount usability engineering*”, é um método de baixo custo, rápido e fácil de usar (Nielsen, 1993). É uma técnica analítica e informal proposta por Nielsen e Molich (1990). Consiste em ter de 3 a 5 avaliadores analisando a interface segundo um conjunto de dez heurísticas de usabilidade. Cada avaliador deve analisar a interface de maneira independente, produzindo um relatório contendo os problemas identificados. Em seguida, o grupo se reúne para produzir um relatório único contendo os problemas identificados por todos os avaliadores. Para cada problema é atribuído um grau de severidade, variando de 1 a 5, que representa a gravidade do problema na opinião do grupo de avaliadores onde 1 equivale ao menor grau de importância e 5 ao maior.

São considerados problemas as situações que infrinjam pelo menos uma das seguintes heurísticas:

1. Visibilidade do *Status* do Sistema

O sistema precisa manter os usuários informados sobre o que está acontecendo, fornecendo um *feedback* adequado dentro de um tempo razoável

2. Coerência do Sistema com o Mundo Real



O sistema precisa falar a língua do usuário, com palavras, frases e conceitos familiares ao usuário, ao invés de termos orientados ao sistema. Deve seguir convenções do mundo real, fazendo com que a informação apareça numa ordem natural e lógica

3. Controle e Liberdade do Usuário

Os usuários freqüentemente escolhem por engano funções do sistema e precisam ter claras saídas de emergência, para sair do estado indesejado sem ter que percorrer um extenso diálogo.

4. Consistência e Padronização

Os usuários não devem ter que adivinhar que diferentes palavras, situações ou ações significam a mesma coisa. Seguir convenções de plataforma computacional (de acordo com o público alvo)

5. Prevenção de Erros

Melhor que uma boa mensagem de erro é um design cuidadoso, que previne o erro antes que aconteça.

6. Reconhecer é melhor que relembrar

O sistema deve tornar objetos, ações e opções visíveis. O usuário não deve ter que lembrar informação de uma para outra parte do diálogo. Instruções para uso do sistema devem estar visíveis e facilmente recuperáveis quando necessário.

7. Flexibilidade e Eficiência de Uso

Usuários novatos se tornam peritos com o uso. Prover aceleradores de forma a aumentar a velocidade da interação. Permitir a usuários experientes "cortar caminho" em ações frequentes.

8. Estética e *Design* Minimalista

Diálogos não devem conter informação irrelevante ou raramente necessária. Qualquer unidade de informação extra no diálogo irá competir com unidades relevantes de informação e diminuir sua visibilidade relativa.

9. Auxilia no reconhecimento, diagnóstico e recuperação de erros?

Mensagens de erro devem ser expressas em linguagem clara (sem códigos) indicando precisamente o problema e construtivamente sugerindo uma solução.

10. Ajuda e Documentação

Embora seja melhor um sistema que possa ser usado sem documentação, é necessário prover *help* e documentação. Essas informações devem ser fáceis de encontrar, focalizadas na tarefa do usuário e não muito extensas.

Após a identificação dos problemas os avaliadores podem colocar comentários gerais sobre a interface onde é possível descrever problemas não contemplados pelas dez heurísticas.

Em estudos experimentais comparativos com outras técnicas de avaliação como por exemplo o uso de guias de estilo (*guidelines*), percurso cognitivo e testes de usabilidade, a literatura tem mostrado que a avaliação heurística apresenta melhores resultados encontrando problemas mais sérios com menos esforço despendido (Jeffries et al. 1991).

4.1.2 Contexto da aplicação da avaliação heurística

Neste trabalho, a avaliação heurística foi conduzida por quatorze grupos distintos, cada um contendo de 3 a 5 avaliadores. Dessa forma, a interface foi analisada por um número expressivo de pessoas. Os avaliadores cursavam, na época da aplicação dessa técnica, a disciplina de IHC em um curso de especialização na área de computação. Eles estavam portanto praticando esse método de inspeção de usabilidade em situação de aprendizado.

O GECON foi instalado no servidor web da Equipe de EAD da Unicamp. Foi então disponibilizada, para os avaliadores, uma área contendo um site com todos os elementos visíveis porém sem conteúdo algum. Os avaliadores tinham acesso a esse *site* por meio de um usuário com todas as permissões para alterar o *layout* e incluir conteúdo. Dessa forma foi possível aos avaliadores, navegar por toda interface do GECON.

Outro ponto importante a ser destacado é que o GECON estava em desenvolvimento e, portanto, a inspeção foi feita sobre um protótipo inacabado da ferramenta. O fato do sistema estar em desenvolvimento dificultou a tarefa de avaliação, pois alguns problemas identificados simplesmente não existiriam se o desenvolvimento já estivesse concluído. Porém, o fato do sistema não estar concluído, permitiu a correção de muitos problemas de usabilidade antes mesmo do teste piloto de uso.

4.1.3 Resultados das avaliações

As 14 avaliações foram analisadas gerando-se uma tabela para cada uma das 10 heurísticas. Muitas vezes um mesmo problema é mencionado de formas diferentes pelos

grupos. Foi feita uma interpretação para síntese de problemas identificados por mais de um grupo.

Os problemas apontados por apenas um grupo foram agrupados em uma mesma linha da tabela sob o título de “Outros”.

Essa forma de organização deixa evidente que determinados problemas são apontados por um único grupo ao passo que outros são apontados por vários grupos. Com isso é possível identificar quais são os problemas mais aparentes simplesmente pelo número de grupos que os apontaram.

Para facilitar a visualização dos problemas foram gerados gráficos de barras onde no eixo x são colocados os problemas e no eixo y o número de grupos que os apontou. Para simplificar os gráficos enfatizando os problemas mais relevantes, todos os problemas apontados por apenas um grupo (menor relevância) foram reunidos e representados por uma mesma barra denominada “Outros” (os outros problemas).

As 14 avaliações heurísticas, tais como foram geradas pelos grupos, encontram-se no apêndice A.

Heurística 1: Visibilidade do status do sistema

Foram identificados 18 pontos relativos a essa heurística que são apresentados no gráfico a seguir (Figura 4.1).

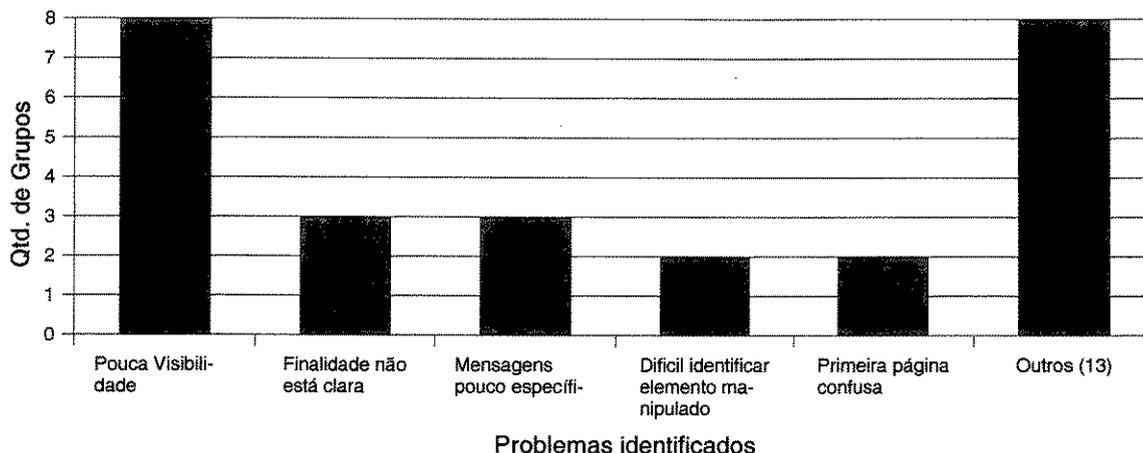


Figura 4.1 – Gráfico da Heurística Visibilidade

Oito dos quatorze grupos apontaram a falta de visibilidade do sistema, ou seja, a resposta do sistema às ações do usuário não era suficientemente clara para o usuário concluir se o sistema havia executado a tarefa desejada ou não.

Outro ponto identificado foi a falta de clareza quanto a finalidade do sistema. Dois grupos disseram que a primeira página é muito confusa, o que reflete o fato do sistema possuir pouca informação sobre seu funcionamento. Configurou-se assim um aspecto a ser melhorado no sistema.

Outro questionamento feito por 3 grupos diz respeito às mensagens de *feedback* pouco específicas, o que sugeriu que elas deviam ser tratadas com mais cuidado quanto a sua semântica.

A dificuldade de identificação de qual elemento estava sendo manipulado foi um problema apontado por dois outros grupos. Um terceiro grupo apontou falta de clareza nos

menus. Esses dois problemas indicavam que o menu precisava ser melhorado fornecendo a informação de qual elemento estava sendo manipulado.

No geral pôde-se concluir que o aspecto visibilidade precisava ser melhor trabalhado.

Heurística 2: Coerência do sistema com o mundo real

Foram identificados 14 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.2).

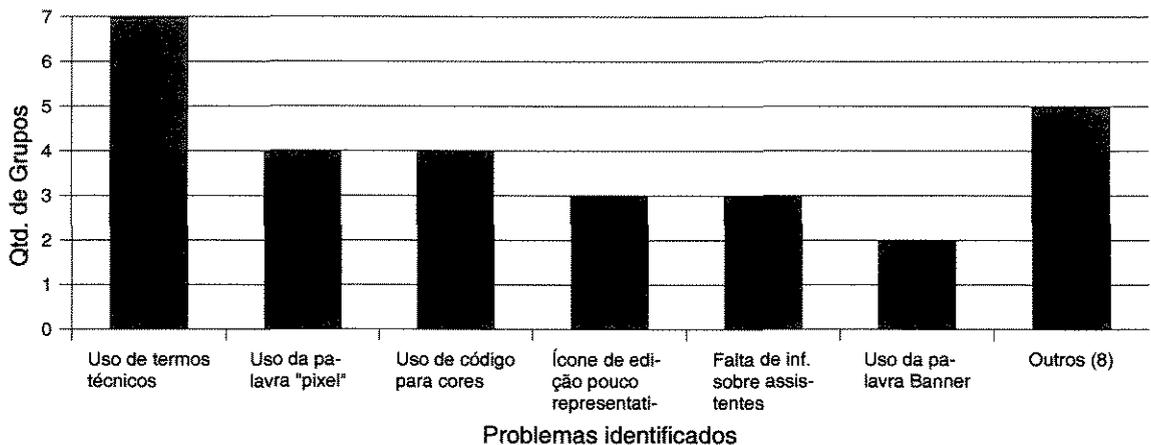


Figura 4.2 – Gráfico da Heurística Coerência com o mundo real

Em comparação com a heurística de visibilidade, mais grupos identificaram os mesmos pontos. O grande problema do sistema quanto a coerência com o mundo real segundo a maioria dos grupos foi o uso de termos técnicos.

Entre os sete grupos que apontaram a existência de termos técnicos, quatro especificaram a palavra “*pixels*” como exemplo e outros dois grupos deram como exemplo de termo técnico a palavra “*banner*”.

Quatro grupos mencionaram que o uso de códigos para definir as cores era um problema de incompatibilidade com o mundo real.

Três grupos disseram que o símbolo gráfico usado para abrir a janela administrativa era pouco representativo.

Também não estava claro para o usuário o que significavam os assistentes na área principal. Esse ponto foi abordado também por três grupos.

Heurística 3: Controle e Liberdade

Foram identificados 15 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.3).

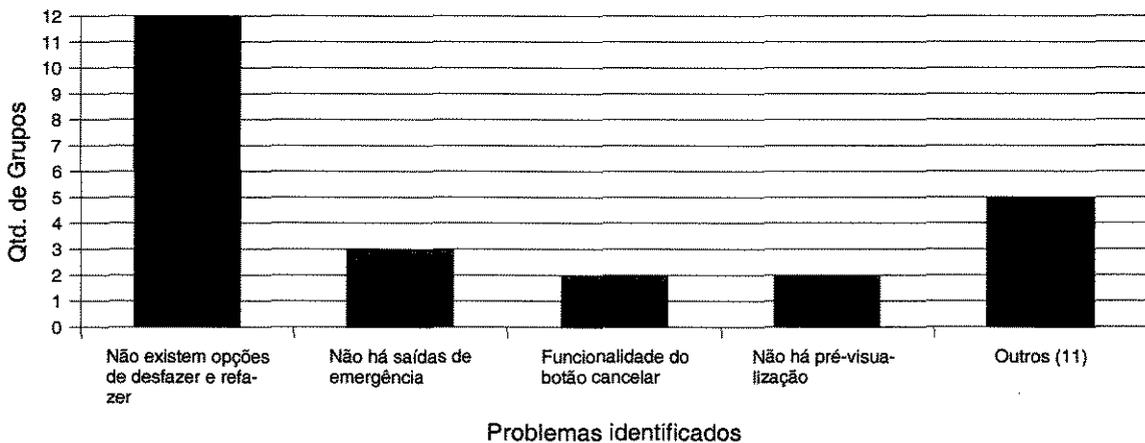


Figura 4.3 - Gráfico da Heurística Controle e Liberdade

Doze dos quatorze grupos mencionaram a ausência de opções de desfazer e refazer como problema. Um outro problema, mencionado por três grupos, foi a ausência de saídas de emergência.

Um ponto interessante que foi comentado por dois grupos é que o sistema não provia mecanismos para visualizar uma alteração antes dela ser efetivamente disponibilizada no *site*.

Heurística 4: Consistência e Padronização

Foram identificados 23 pontos relacionados a essa heurística, comentados e ilustrados na Figura 4.4, a seguir.

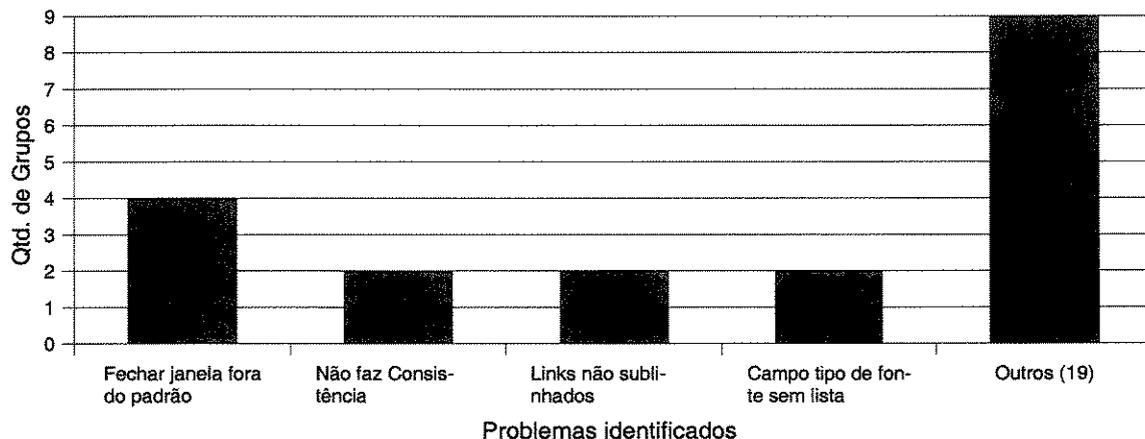


Figura 4.4 - Gráfico da Heurística Consistência e Padronização

Esta foi a heurística que, em números absolutos, apresentou o maior número de pontos identificados. Apesar disso o problema mais aparente foi apontado por apenas quatro grupos. O fato da opção fechar janela estar em forma de *link*, não seguindo o padrão das outras duas opções “confirmar” e “cancelar” que são apresentadas como botões, mostrou-se um problema.

Apenas outros três pontos foram levantados por mais de um grupo. O fato de não ser feita nenhuma verificação nos dados informados nos formulários foi enumerado por dois grupos. Os outros dois pontos foram: os *links* não apareciam sublinhados e o campo de tipo de fonte não apresentava a lista de fontes disponíveis. Outros dezenove problemas eram diversos e foram apontados por apenas 1 grupo cada.

Heurística 5: Prevenção de Erros

Foram identificados 14 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.5).

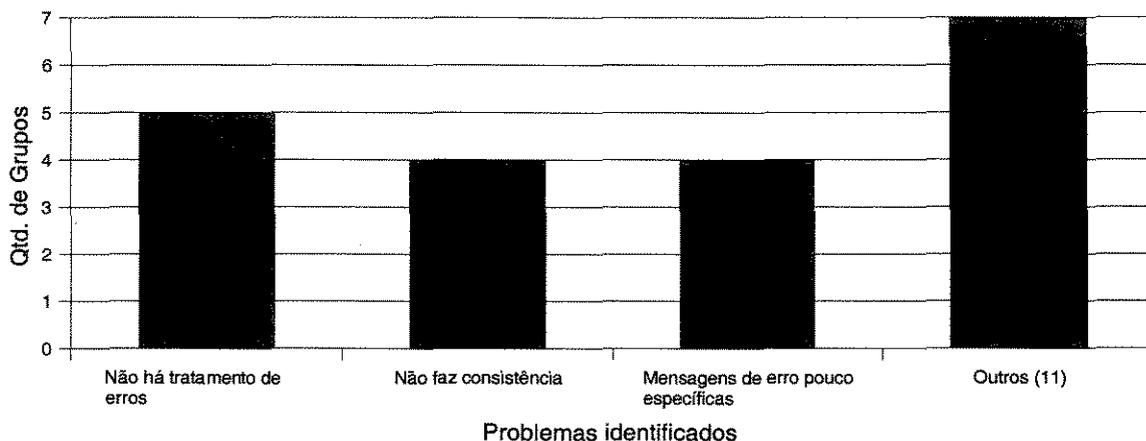


Figura 4.5 - Gráfico da Heurística Prevenção de Erro

Dos quatorze problemas apontados apenas três foram citados por mais de um grupo. O primeiro foi a falta de tratamento de erros apontado por cinco grupos e o segundo foi a falta de validação das informações preenchidas pelos usuários. De fato, esses dois pontos ainda não haviam sido tratados no estágio de desenvolvimento em que estava o sistema

quando foram feitas as inspeções. O terceiro ponto refere-se às mensagens de erros serem pouco específicas, o que levou à revisão de todas as mensagens procurando melhorar a qualidade da informação contida nas mesmas.

Heurística 6: Reconhecer é melhor que relembrar

Foram identificados 7 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.6).

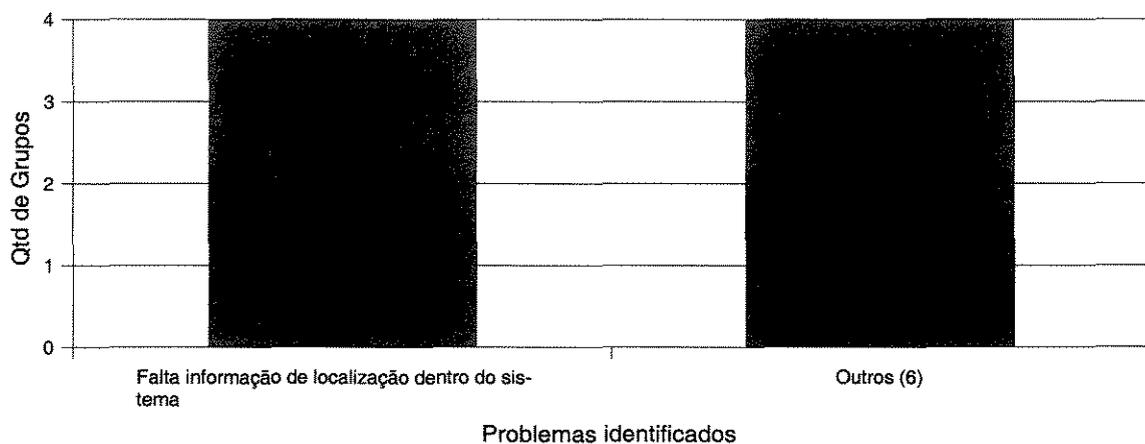


Figura 4.6 - Gráfico da Heurística Reconhecer X Relembrar

Para essa heurística foi apontado o menor número de problemas. Apenas quatro grupos mencionaram de alguma forma que era difícil saber em que parte do sistema estavam. Outros quatro grupos identificaram seis outros problemas.

Um dos aspectos citados, deve ser esclarecido. Foi mencionado por um dos grupos que: *“deveria explorar mais o conceito de frames para facilitar ao usuário a se localizar dentro do sistema, por exemplo, o menu principal poderia ficar fixo e não rolando junto*

com a página;”. Segundo Nielsen (Nielsen, 2000) o uso de *frames* provoca outros problemas de usabilidade; por isso optou-se por não utilizá-los. Para minimizar o problema do menu principal de rolar para fora do campo de visão do usuário, procurou-se reorganizar as páginas, deixando-as curtas o suficiente para que coubessem em uma única tela na resolução 800X600 (resolução mais comum na época em que este trabalho foi escrito).

Outro ponto comentado foi: “os submenus estão confusos, deveria ser dado um destaque para a opção ativada.” Essa questão dos menus também foi mencionada na heurística visibilidade por outro grupo. Isso foi resolvido usando um recurso de CSS (Cascade Style Sheet) para ressaltar a opção ativada tanto no menu principal quanto nos secundários dando sempre ao usuário uma indicação de onde ele está.

Heurística 7: Flexibilidade e Eficiência de Uso

Foram identificados 17 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.7).

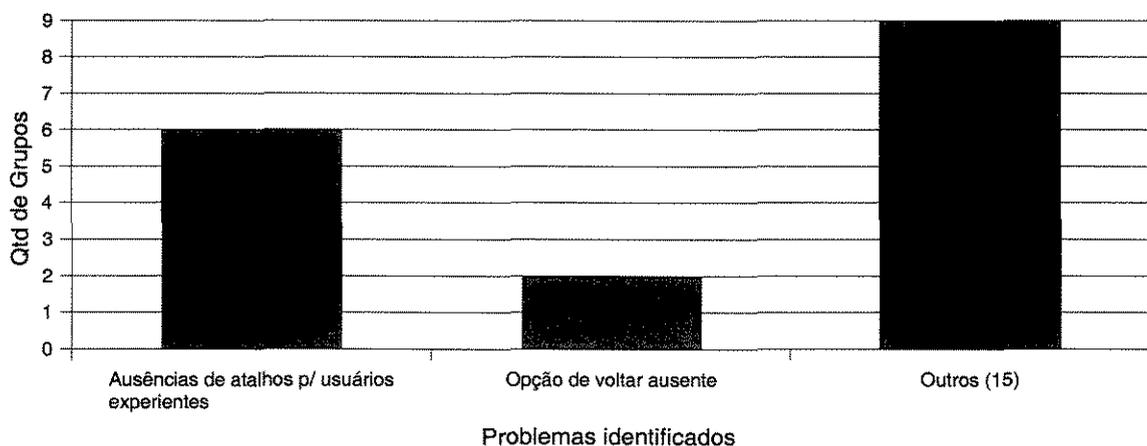


Figura 4.7 - Gráfico da Heurística Flexibilidade e Eficiência de Uso

O ponto principal apontado por seis grupos foi a ausência de atalhos específicos para usuários experientes; porém, nenhum grupo mencionou qual atalho teria sido interessante. Outro ponto identificado por mais de um grupo é que não existia a opção *Voltar* pois o botão *Cancelar* fechava a janela ao invés de voltar para a ação anterior.

Na versão seguinte do sistema todos os botões ao final das telas administrativas foram redefinidos. O botão “cancela” foi substituído por um botão “voltar” com a funcionalidade de voltar à página anterior e a opção “fechar janela” foi substituída por um botão “sair” a direita dos outros dois.

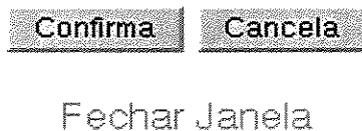


Figura 4.8- Botões antes da avaliação heurística



Figura 4.9 - Botões após a redefinição

Heurística 8: Estética e Design Suficiente

Foram identificados 18 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.10).

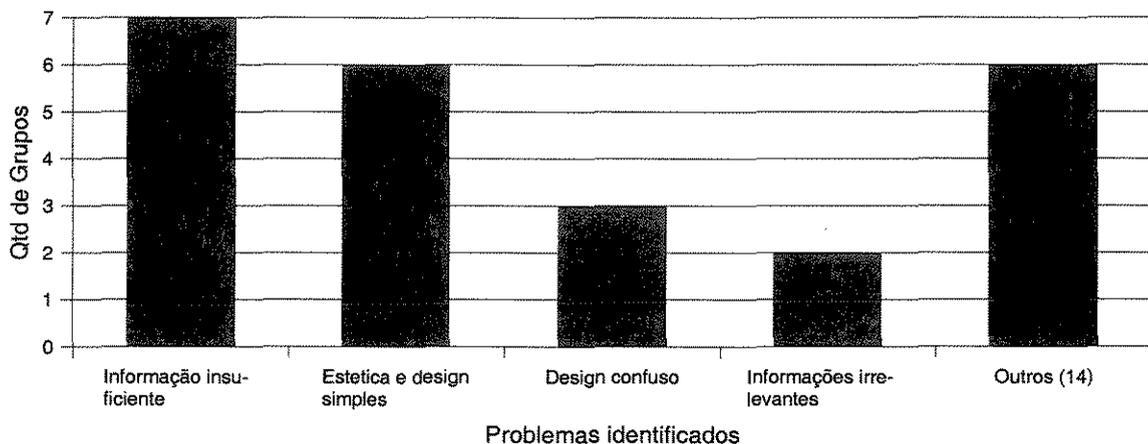


Figura 4.10 - Gráfico da Heurística Estética e Design Suficiente

A metade dos grupos mencionou que a informação contida na interface era insuficiente para o usuário utilizar o sistema e seis grupos disseram que a estética e o design eram excessivamente simples. A simplicidade foi proposital pois ela é uma premissa básica para uma boa usabilidade. Porém era necessário prover mais informação ao usuário sobre o funcionamento do sistema.

Três grupos disseram que o design era confuso e dois disseram que o sistema possuía informação irrelevante sem, no entanto, informar qual seria.

Heurística 9: Auxilia no reconhecimento, diagnóstico e recuperação de erros?

Foram identificados 10 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.11).

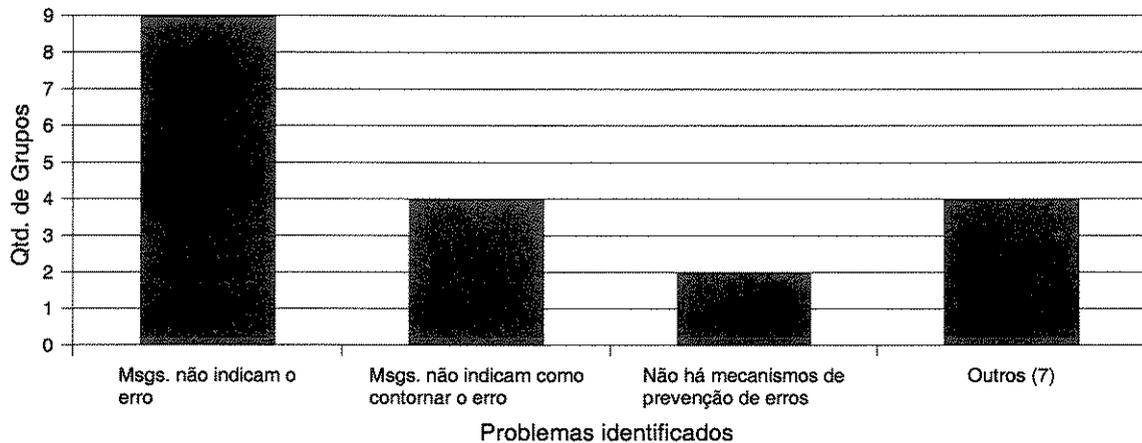


Figura 4.11 - Gráfico da Heurística Recuperação de Erros

Dez problemas foram apontados pelos quatorze grupos. O mais evidente apontado por nove grupos era que as mensagens de erro não indicavam qual tinha sido o erro e o segundo ponto descrito por quatro grupos indicavam que as mensagens de erro não mostravam como contornar o erro. Porém, apenas dois grupos apontaram o principal problema relacionado a esta heurística, que é o fato de simplesmente não existir mecanismo de prevenção de erro implementado no momento em que foram feitas as avaliações heurísticas. Sendo assim, as mensagens de erro apresentadas eram as do Zope e não do GECON.

Heurística 10: Ajuda e Documentação

Foram identificados 5 pontos relacionados a essa heurística que são apresentados no gráfico a seguir (Figura 4.12).

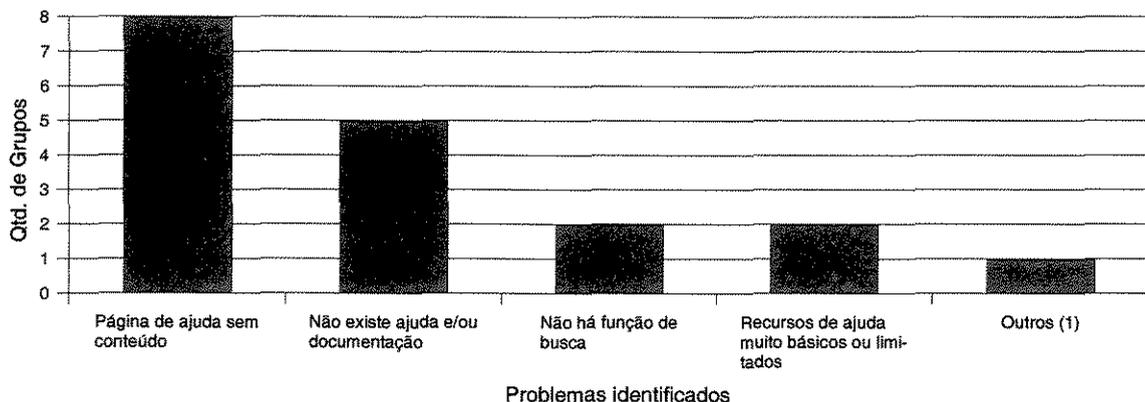


Figura 4.12 - Gráfico da Heurística Ajuda e Documentação

Essa heurística, foi a que apresentou o menor número de problemas identificados.

O ponto mais evidente foi que a opção de ajuda estava sem conteúdo. Esse fato foi apontado por dez grupos. Cinco grupos constataram que de fato não havia ajuda ou documentação. A opção de ajuda estava prevista porém não havia sido implementada quando esta avaliação foi feita.

O terceiro ponto mais evidente apontado por apenas dois grupos foi que não existia opção de busca.

4.1.4 Discussão geral sobre os resultados da Avaliação Heurística

O fato da avaliação heurística ter sido feita durante a fase de desenvolvimento do GECON, contribuiu para o uso efetivo da primeira versão do sistema durante o teste piloto. Isto porque muitos pontos identificados na avaliação, foram corrigidos na interface antes da realização do teste.

Praticamente todos os pontos levantados durante a avaliação heurística foram retrabalhados à exceção da opção de desfazer e refazer, por se tratar de uma alteração complexa que demandaria um tempo maior que o disponível.

Como foram feitas muitas alterações na interface do GECON, em decorrência da avaliação heurística, nas duas figuras a seguir são ilustrados apenas alguns exemplos de alterações significativas na interface realizadas com base nos resultados da aplicação desse método. Na primeira (Figura 4.13) não havia sido corrigido o ajuste na tela evitando a barra de rolagem vertical. Também havia o problema da localização dentro do sistema. Inicialmente, havia uma frase que tinha o objetivo de informar ao usuário sua localização no sistema; “Opções para o bloco parcerias” como pode ser visto na Figura 4.13.

Com a alteração da cor das opções do menu, dando destaque à opção selecionada, ficou bem mais fácil para o usuário localizar-se no sistema. Esse fato tornou a frase desnecessária. A Figura 4.14 mostra a tela de inclusão de parcerias, remodelada.

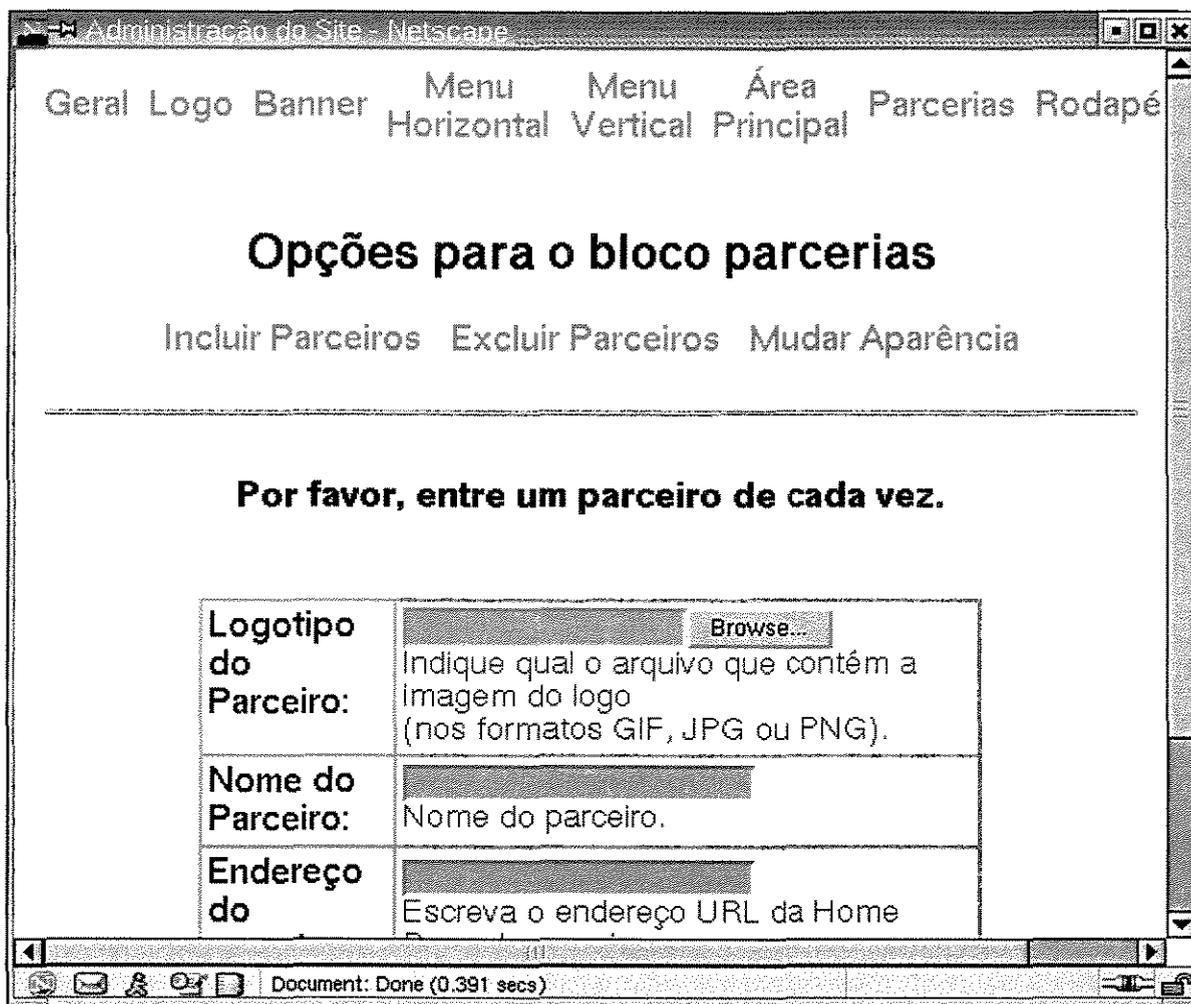


Figura 4.13 - Tela de inclusão do elemento parcerias

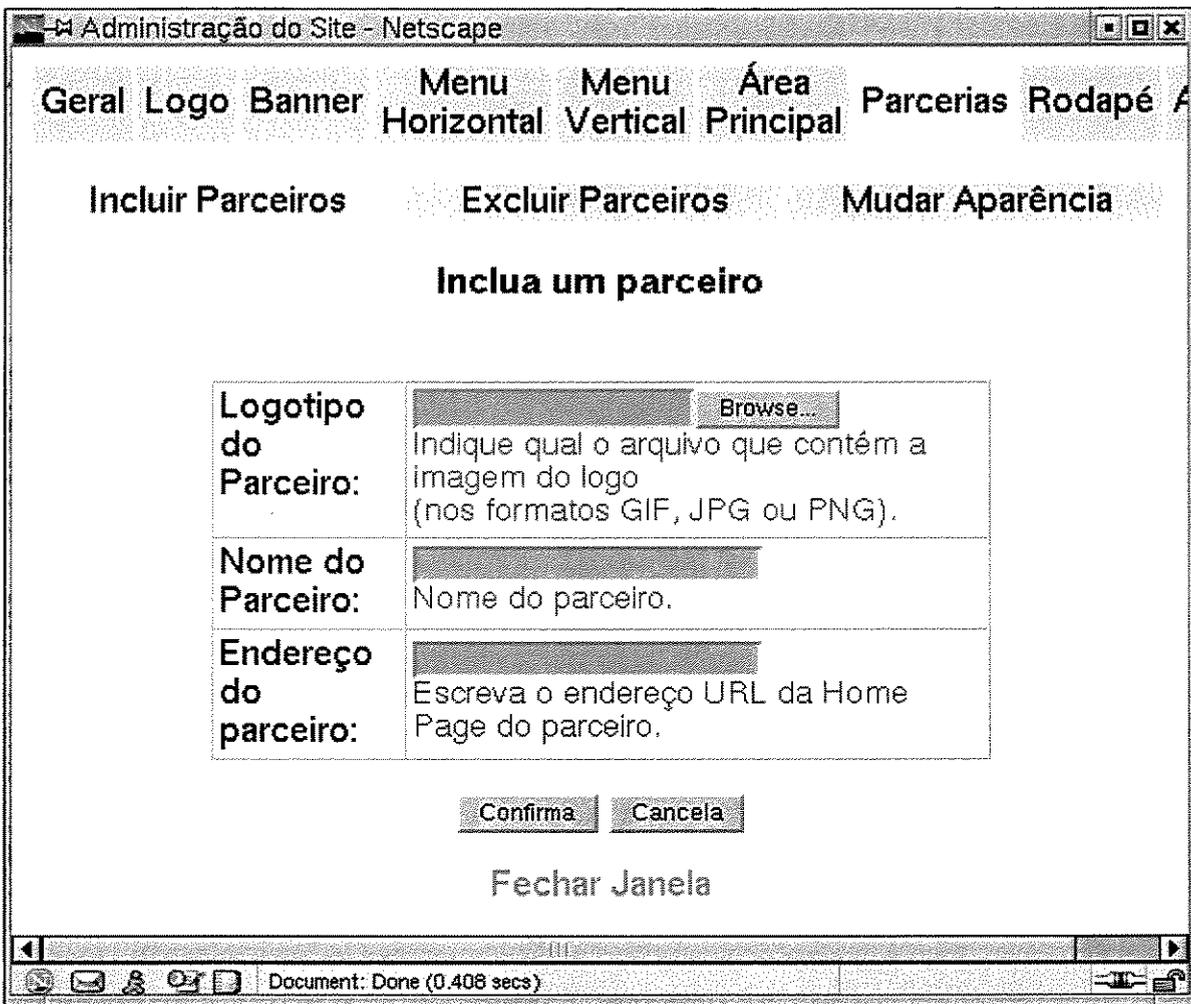


Figura 4.14 - Tela de inclusão do elemento parcerias remodelada

No gráfico da Figura 4.15 estão agrupados os resultados dos problemas levantados em todas as heurísticas. São utilizadas duas barras para representar cada uma delas. A primeira informa a quantidade de problemas considerados mais relevantes por terem sido apontados por mais de um grupo e a segunda barra indica a quantidade total de problemas.

A variação do total de problemas de uma heurística a outra foi de dezoito, sendo vinte e três na heurística consistência e padronização e cinco na ajuda e documentação. Porém, se considerarmos apenas os problemas mais relevantes, a variação é de apenas quatro onde a heurística coerência do sistema com o mundo real apresentou o número máximo de seis enquanto o mínimo de dois foi registrado para flexibilidade e eficiência de uso.

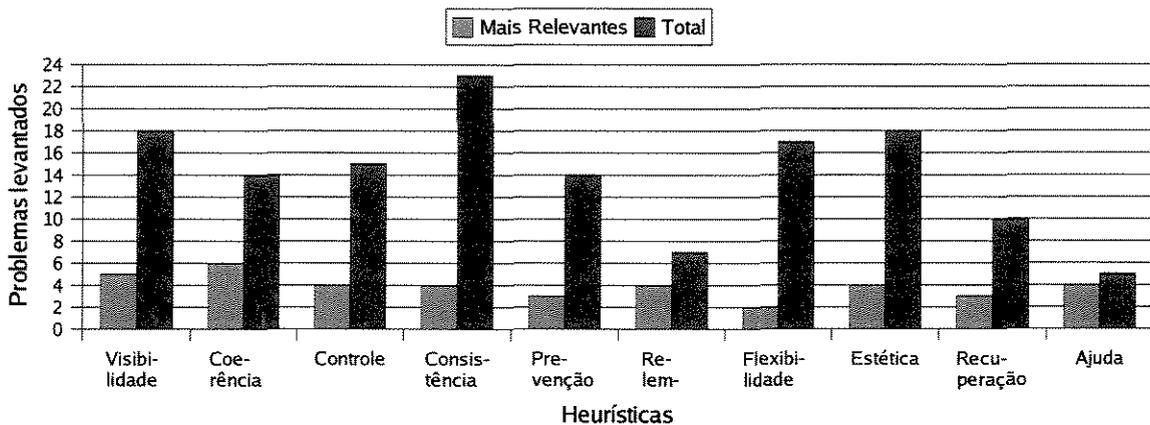


Figura 4.15 - Gráfico dos Problemas por Heurística

A grande diferença numérica entre o total de problemas e os mais relevantes mostra que mesmo nas heurísticas onde foram apontados muitos problemas, o número de problemas relevantes é bastante reduzido.

O gráfico deixa claro também que foram encontrados problemas em todas as heurísticas indicando que a avaliação foi abrangente.

O resultado geral representado na Figura 4.15 mostra que a avaliação heurística mostrou-se efetiva como instrumento para indicar melhorias ao longo dos diversos aspectos captados pelas 10 heurísticas.

Embora o conjunto dos grupos de avaliadores estivessem em processo de aprendizagem, os resultados obtidos puderam ser validados e forneceram subsídio para melhorias.

4.2 Avaliação Piloto de Uso do Sistema

Após a reformulação da interface do sistema com base no levantamento decorrente da avaliação heurística foi feita uma avaliação piloto de uso do GECON.

A seguir é feita uma descrição do contexto no qual foi realizada essa avaliação e os resultados obtidos.

4.2.1 Contexto da avaliação piloto

A avaliação piloto foi realizada durante os meses de maio e junho de 2003 e envolveu a criação do *site* de EAD (Educação a Distância) da Unicamp (Figura 4.16) com o GECON. Participaram desta avaliação, todos os integrantes da equipe de EAD da Unicamp colaborando com críticas e sugestões de melhorias. Ao todo, foram sete pessoas com diferentes perfis profissionais sendo um estagiário, um técnico em processamento de dados e cinco analistas de sistemas. O trabalho foi distribuído de tal forma que cada sessão do *site* ficou sob a responsabilidade de uma pessoa.

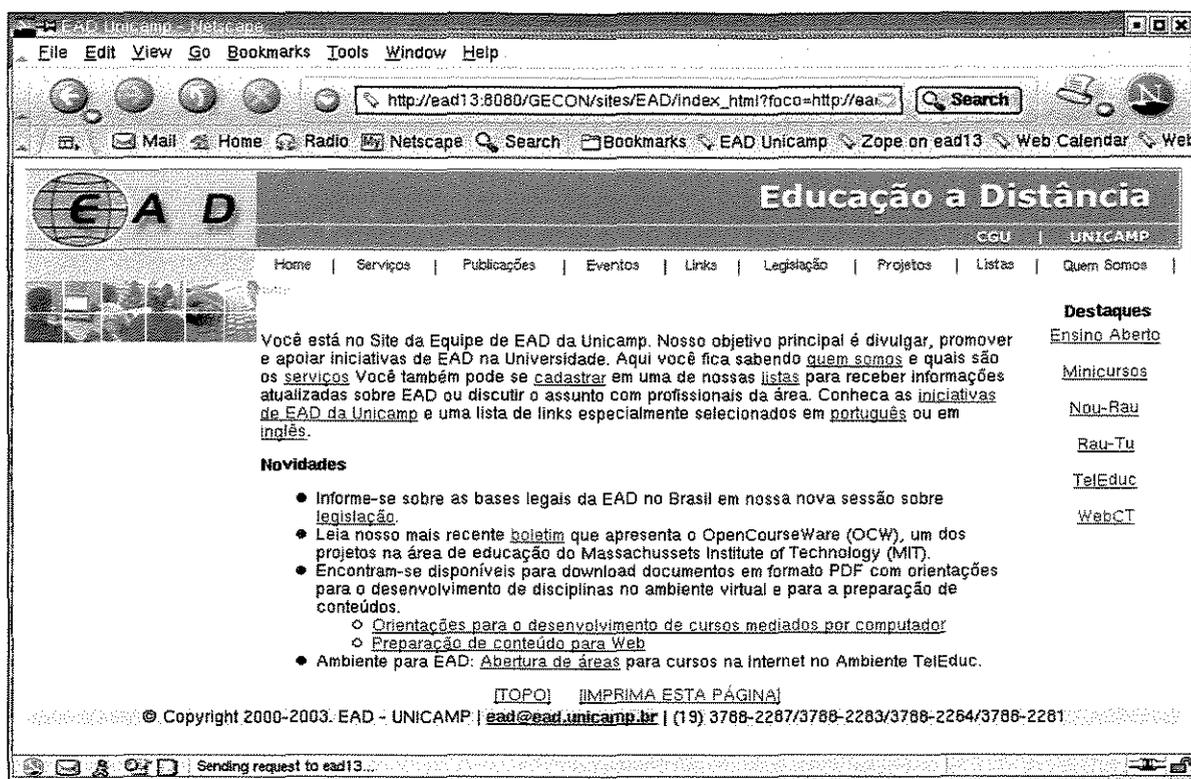


Figura 4.16 - site de EAD feito com GECON

As seções criadas foram: Serviços, Publicações, Eventos, Links, Legislação, Projetos, Listas e Quem Somos.

Para a criação das seções foram usados os assistentes da opção área principal do GECON. A parte de *Eventos*, *Links* e *Publicações* possuía cada uma delas, um assistente próprio e as demais seções foram criadas por meio do assistente de HTML simples. Os assistentes apresentam formulários nos quais são inseridas informações para gerar as páginas. O assistente de HTML diferencia-se dos demais por não possuir uma formatação pré-definida das páginas a serem geradas.

Para cada pessoa foi realizada uma breve apresentação do GECON e do assistente que seria utilizado por ela. Durante a apresentação foram discutidas dúvidas e sugestões de alterações para o GECON que estão descritas na próxima seção desse capítulo.

4.2.2 Resultados da avaliação piloto

Quase todas as pessoas tiveram dúvidas em relação ao processo de criação de conteúdo novo (descrito na seção “Elemento Área Principal” do capítulo 3), o que sugere a existência de um problema de usabilidade da interface em relação ao acesso a essa funcionalidade.

Para iniciar a criação de um novo conteúdo é preciso clicar sobre o ícone administrativo ao lado da área principal para acessar a janela administrativa desse elemento. Como a área principal está sempre exibindo algum conteúdo, as pessoas relataram que tinham a impressão de que clicando no ícone ao lado daquele conteúdo o sistema iria permitir apenas alterá-lo e não criar um conteúdo novo.

Uma sugestão feita durante o teste piloto de uso para minimizar esse problema foi a incorporação de um texto explicativo ao lado dos ícones conforme ilustrado na Figura 4.17 a seguir.

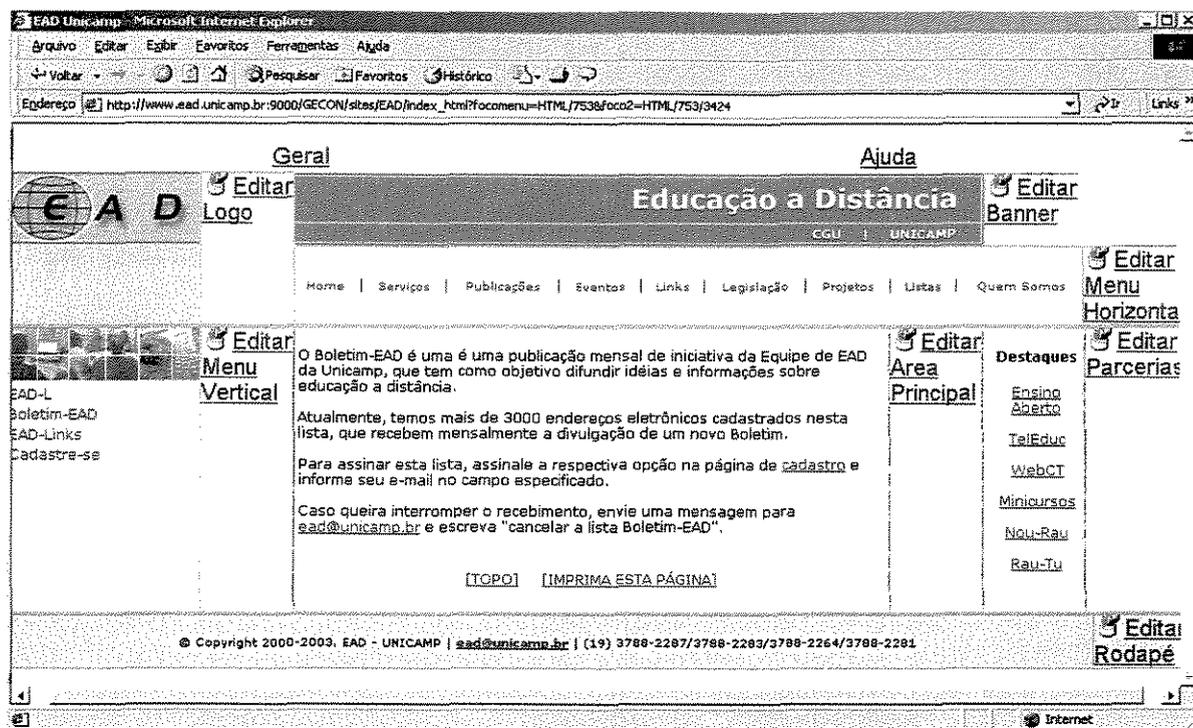


Figura 4.17 – Interface administrativa com texto explicativo junto aos ícones

A seguir são comentados resultados de utilização dos diversos assistentes para criação de conteúdos, disponibilizados pelo sistema.

Trabalho com o assistente de eventos

O assistente de eventos, assim como os demais, funciona da seguinte forma: os eventos são cadastrados por meio de um formulário (Figura 4.18) no qual são preenchidas as informações básicas sobre o evento (nome, datas de início e fim, responsável, URL do site do evento e um campo ementa para criação de uma página sobre o evento). Os eventos são organizados em categorias que são exibidas no site por meio de um link no menu vertical (Unicamp, Brasil, Mundo, no lado esquerdo da Figura 4.19). Ao clicar sobre esse

link, a área principal exibe uma listagem dos eventos cadastrados naquela categoria (Figura 4.19) A listagem contém, para cada evento, informações básicas e um *link* para a página criada no campo ementa, contendo informações mais detalhadas relativas a ele (Figura 4.20). Nessa página é apresentado um *link* para a página do evento.

The screenshot shows a web browser window titled 'Administração do Site - Netscape'. The browser's menu bar includes 'Geral', 'Logo', 'Banner', 'Menu Horizontal', 'Menu Vertical', 'Área Principal', 'Parcerias', 'Rodapé', and 'Ajuda'. Below the menu bar, there are four tabs: 'Assistente de eventos (Eventos)', 'Assistente para páginas html simples (HTML)', 'Assistente para Links (Links)', and 'Assistente de Publicações (Publicações)'. The main content area is titled 'Editando o evento: Tutorial: Gecon - Gerenciador de Sites Dinâmicos'. The form contains the following fields and options:

- Nome do Evento:** Tutorial: Gecon - Gerenciador de Sites Dinâmicos
- Endereço na web:** http://www.ccuec.unicamp.br/treinamentos/tutoriais/programacao/prog_tutoriais.html
- Data de início:** Dia: 26, Mês: Setembro, Ano: 2003
- Data de fim:** Dia: [empty], Mês: [empty], Ano: [empty]
- Pessoa Responsável:** Roberto Romani
- Visível no Menu Vertical:**

Below the form fields, there is a section titled 'Digite a Ementa na área abaixo em um dos três formatos possíveis :'. It includes three radio buttons: 'Texto Simples', 'Texto com HTML', and 'Texto Estruturado'. The 'Texto com HTML' option is selected. The text area contains the following HTML code:

```
<b>DESCRIÇÃO:</b>
<br><br>
Este tutorial apresenta o software GECON para gerenciamento de sites
dinâmicos à distância.
<br><br>
<b>PÚBLICO ALVO:</b>
<br><br>
Professores, monitores e técnicos de suporte em informática.
<br><br>
```

The browser's status bar at the bottom shows 'Document: Done (1.89 secs)'.

Figura 4.18 - Formulário do assistente de eventos

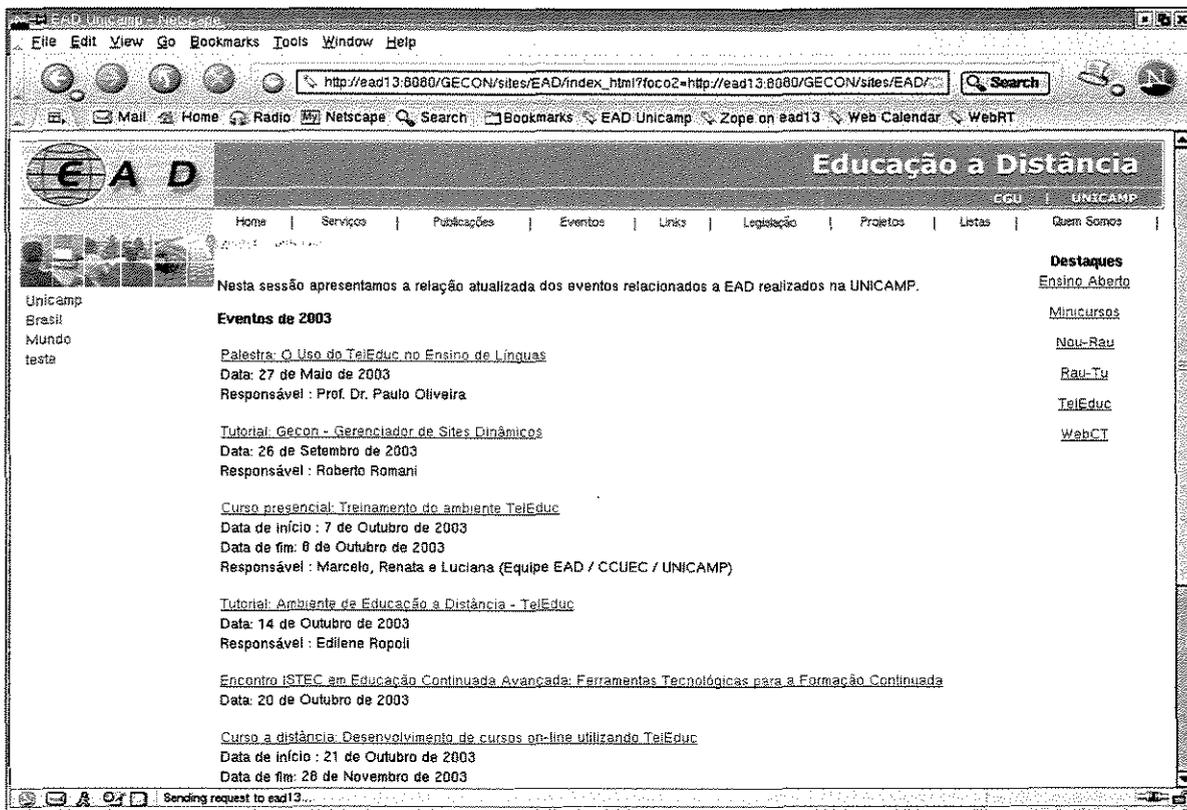


Figura 4.19 - Listagem dos eventos da categoria Unicamp

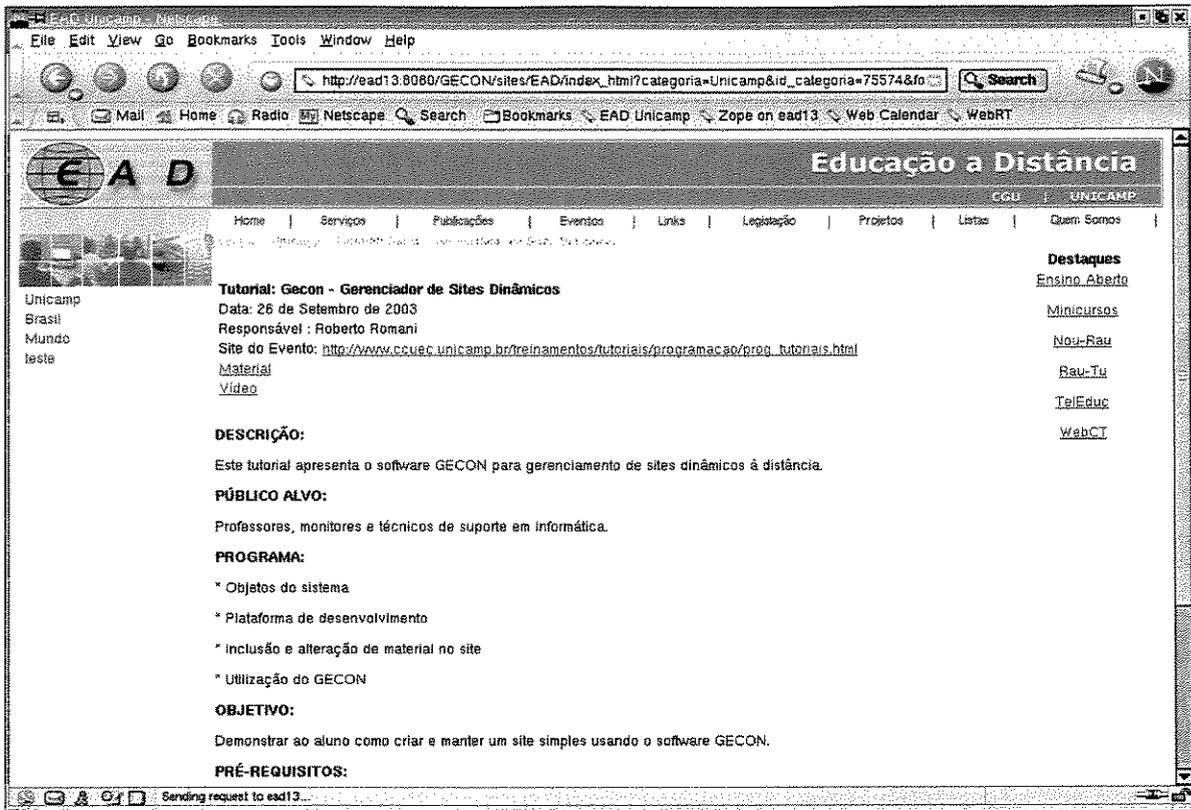


Figura 4.20 - Página gerada a partir do campo ementa

Os pontos identificados durante a discussão sobre o assistente de eventos foram os seguintes.

Mesmo quando não era preenchida informação alguma no formulário sobre o evento, a listagem apresentava o título da informação. Por exemplo o título “data de fim:” era sempre apresentado mesmo que não tivesse sido registrada uma data de fim para o evento.

A *url* do *site* do evento era apresentada na listagem com um *link* para o *site* em questão. Como já existia o *link* para a página do *site* criada via formulário, o *link* para o *site*

do evento foi mantido apenas no cabeçalho da página do evento pois dois *links* juntos na listagem para informações similares poderiam levar à dúvida de onde clicar.

Foi sugerida a definição de um formato fixo para os campos de data. Esses campos eram de preenchimento livre podendo a data ser registrada da forma que o usuário desejasse. Isso implicava na não padronização do formato da data. Com isso, não era permitido que a lista fosse ordenada cronologicamente.

Outro ponto importante identificado foi a necessidade de se fazer uma diferenciação clara entre os eventos passados e os futuros. Como solução foi proposta uma alteração no funcionamento da lista de eventos (Figura 4.19). O funcionamento anterior envolvia apresentar em uma única lista todos os eventos da categoria. A proposta sugerida foi que fossem apresentados na lista apenas os eventos futuros. Porém, a lista deve conter *links* para os eventos passados. Um *link* por ano iniciando sempre pelo ano corrente onde cada um desses *links* apresentaria uma listagem similar a dos eventos futuros só que apenas com os eventos ocorridos no ano correspondente.

Trabalho com o assistente de publicações

A principal sugestão foi a de possibilitar a criação de um texto introdutório para cada categoria de publicação.

Essa sugestão vale também para os assistentes de *links* e eventos. No entanto, sua implementação implicou na revisão da forma de criação de uma categoria pois aumenta a complexidade dessa tarefa.

No início do teste piloto havia apenas um formulário para cada assistente no qual era possível, ao mesmo tempo, criar uma página de conteúdo e uma nova categoria (Figura 4.21).

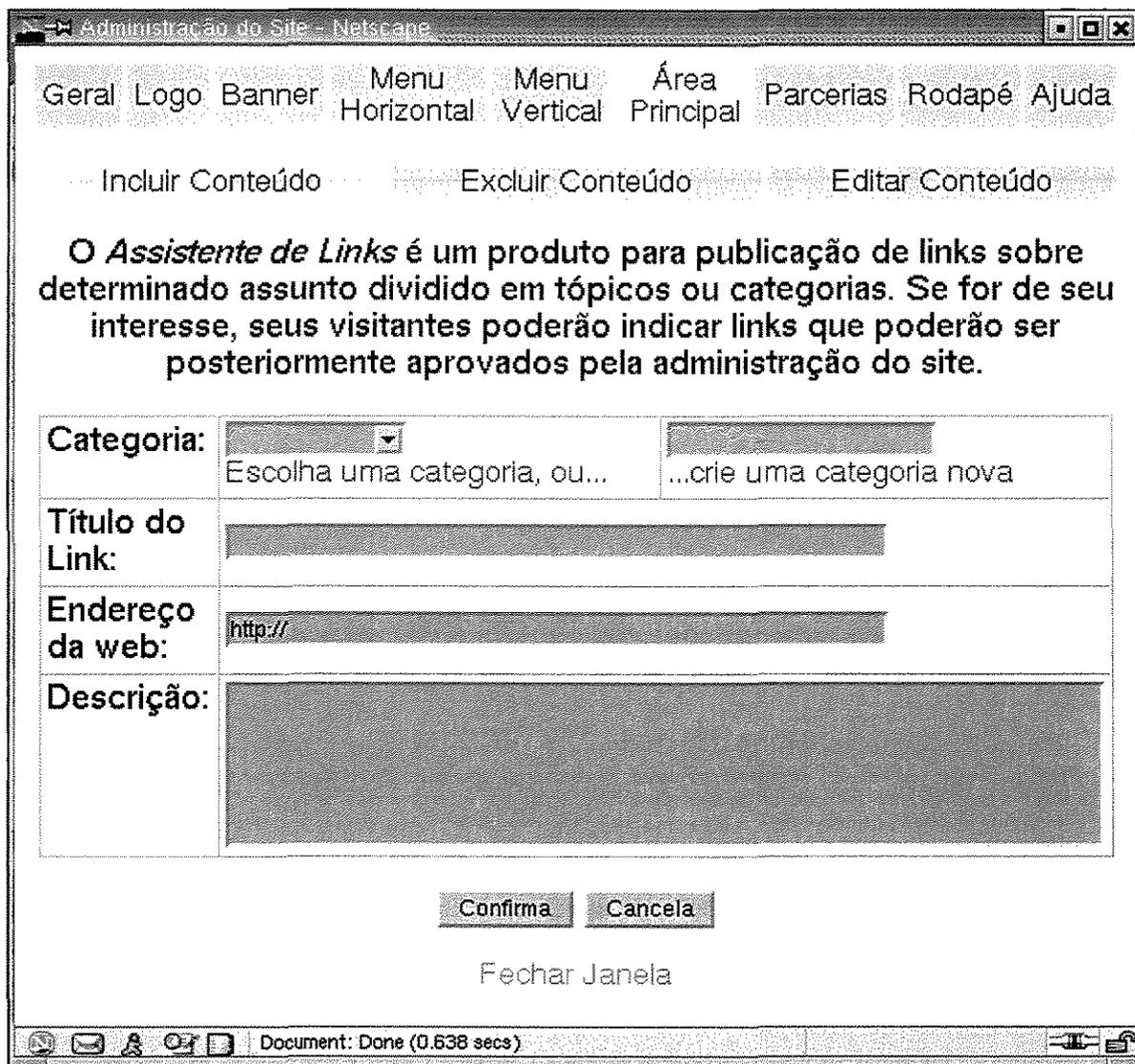


Figura 4.21 - Tela do assistente de *links* antes da separação do formulário de categorias

Para viabilizar a implementação do texto introdutório foi criado um novo formulário para a criação de categorias. Essa nova ação teve que ser incluída no menu do elemento área principal aumentando sua complexidade (Figura 4.22).

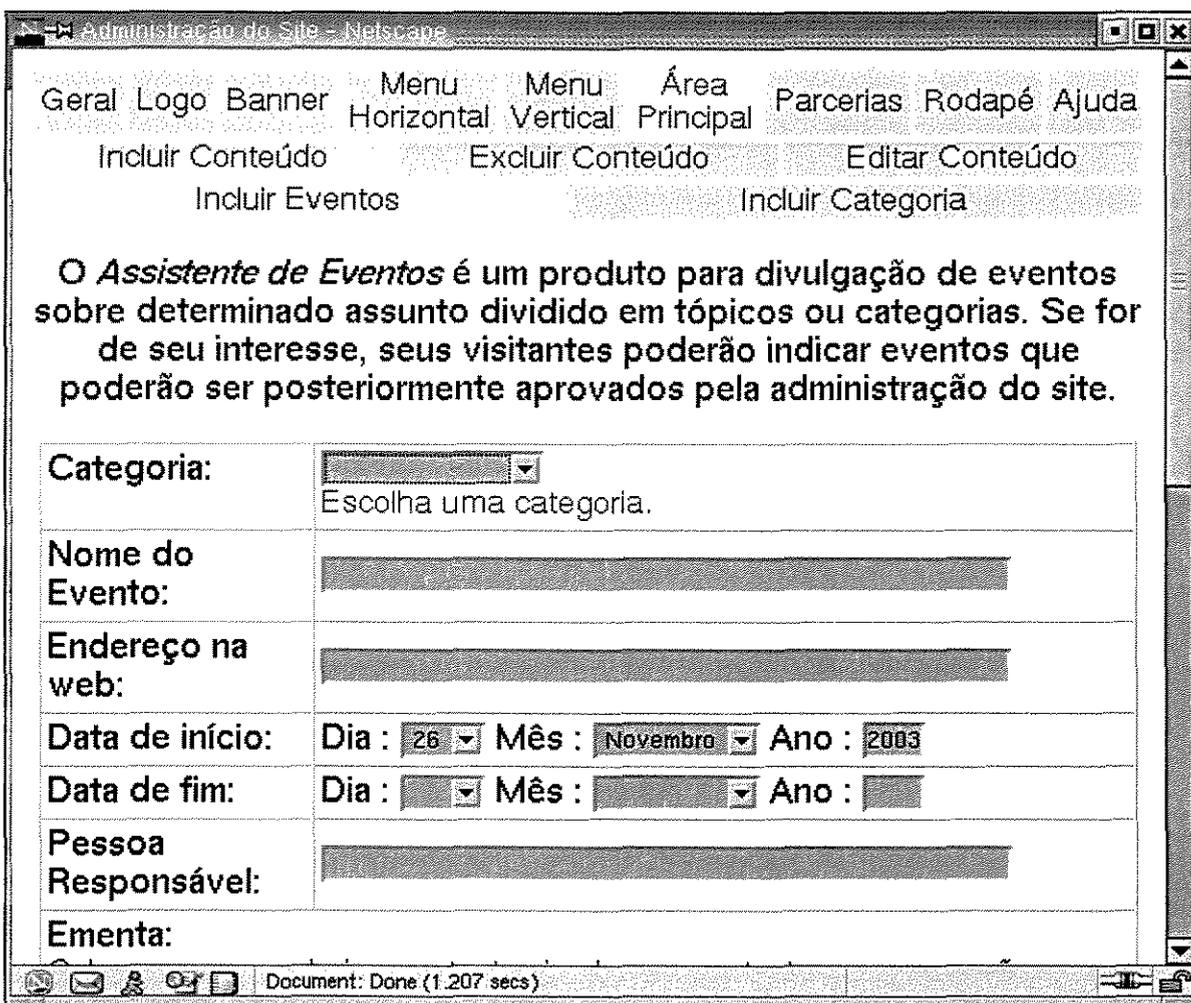


Figura 4.22 - Tela de assistente com submenu para incluir categoria

Outra sugestão dada para o assistente de publicações durante sua apresentação inicial foi a criação de um campo de data no formulário de criação de uma publicação. O objetivo é ordenar a lista das publicações em uma categoria por essa data.

Trabalho com o assistente de Links

Uma sugestão dada para o assistente de *links* foi a de criar subcategorias a fim de organizar melhor a página. É provável que o número de *links* sugeridos cresça muito, dificultando o trabalho do usuário para encontrar o(s) *link(s)* de seu interesse. Isso justificaria a criação de subcategorias.

Outra sugestão foi a de destacar os últimos *links* sugeridos colocando uma sinalização de que aqueles são *links* novos.

Trabalho com o assistente de HTML

No início do teste piloto, para acessar os assistentes era necessário passar por uma tela para selecionar qual assistente seria utilizado (Figura 4.23). Em seguida, era exibida a tela com o formulário para criação ou alteração de conteúdo relacionado ao assistente selecionado.

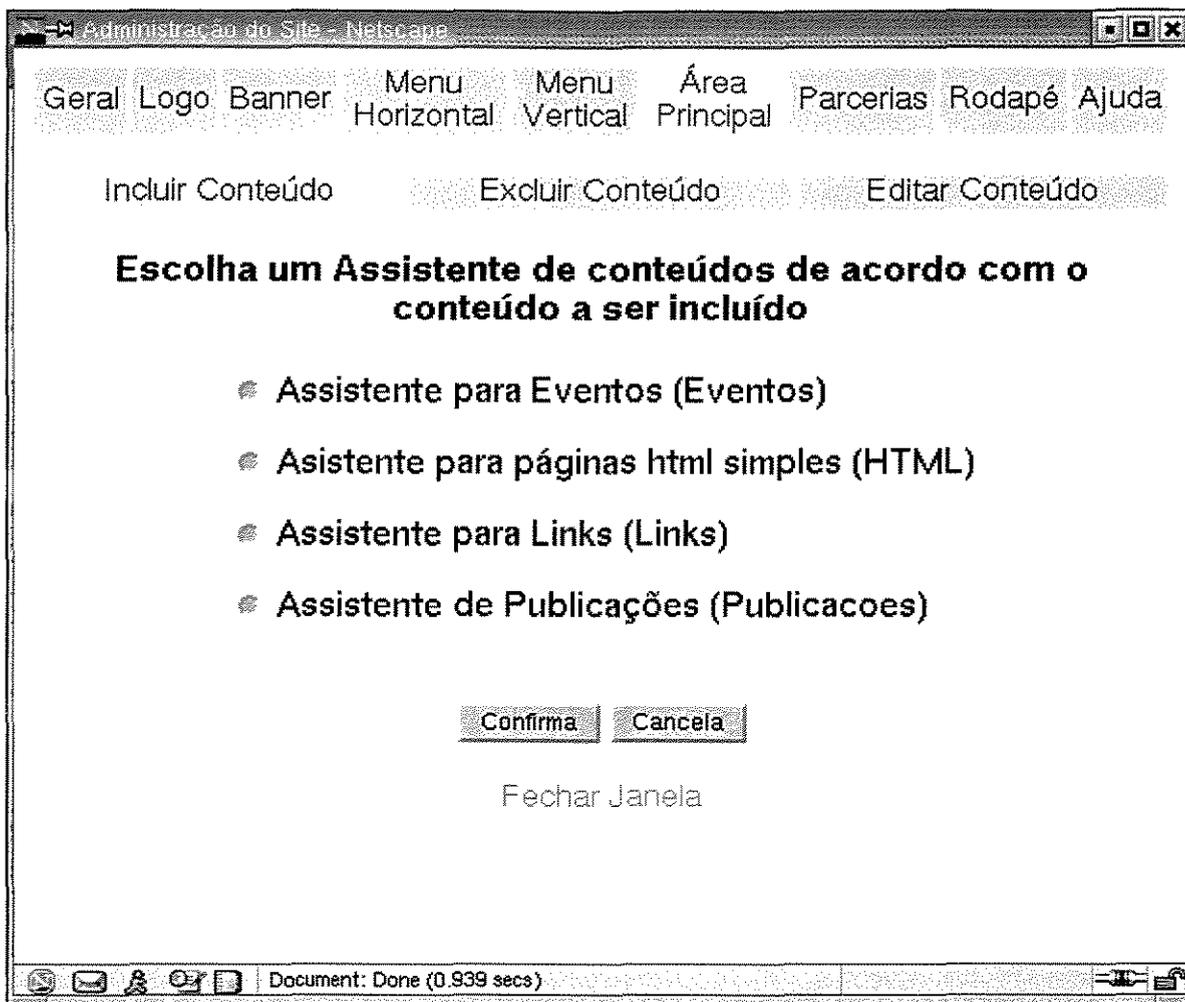


Figura 4.23 - Tela inicial para selecionar assistente

Observou-se que essa escolha poderia ser feita em um submenu do menu área principal (Figura 4.24). Reduzindo o número de cliques e facilitando para o usuário a sua localização no sistema.

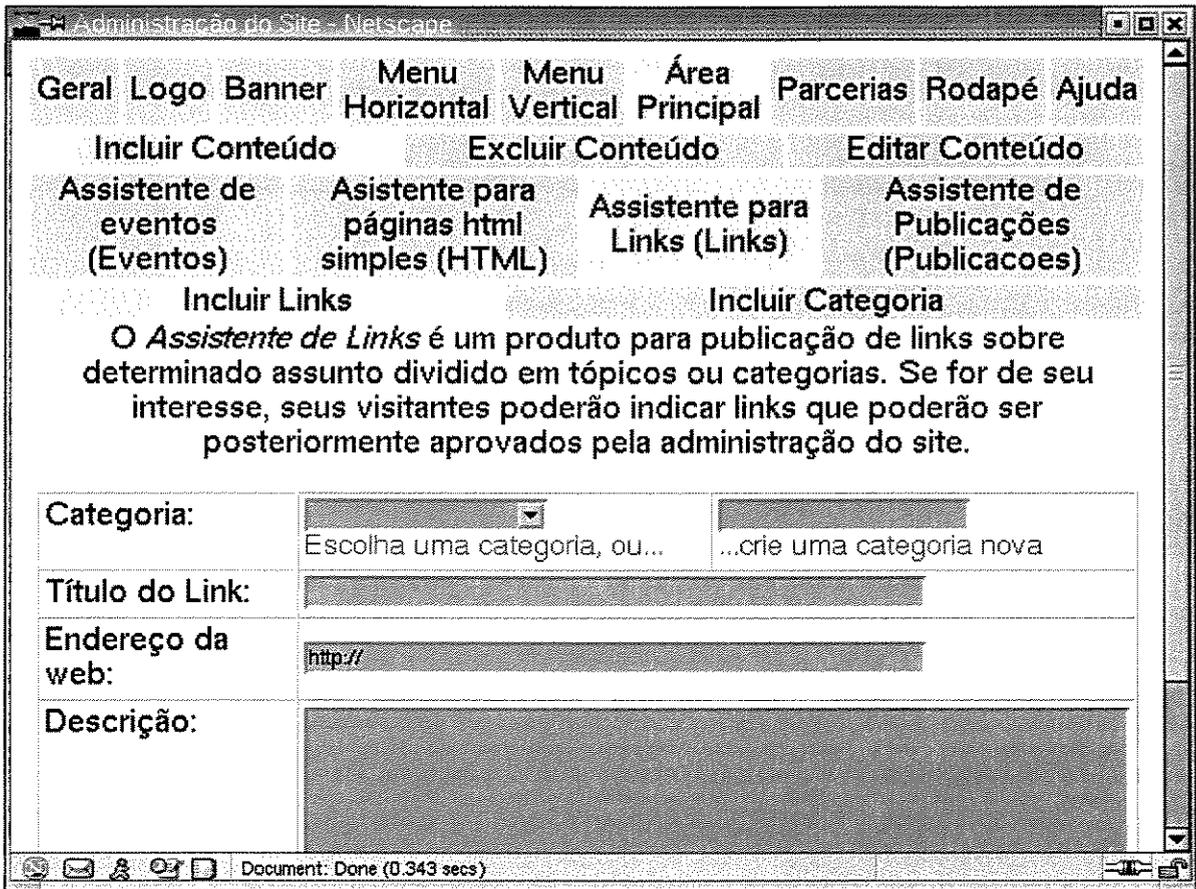


Figura 4.24 - Tela com submenu para selecionar assistente

Porém, essa alteração somada à alteração que incluía um submenu para inclusão de categoria, aumentou muito a complexidade dos submenus do elemento Área Principal. A solução adotada foi, para cada assistente, montar uma estrutura hierárquica navegável contendo as categorias e as páginas de conteúdo com um menu *popup* com as opções de ações específicas para cada caso (Figura 4.25).

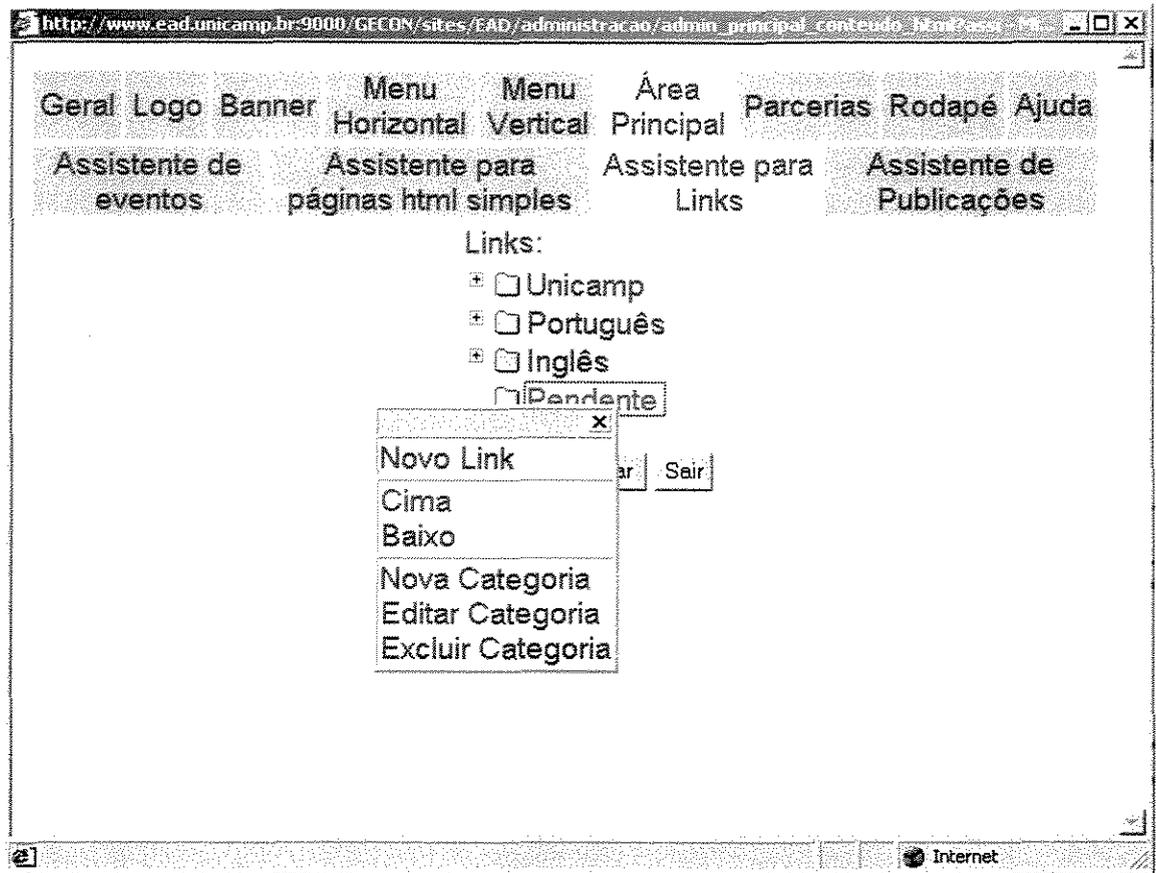


Figura 4.25 - Tela com estrutura hierárquica navegável contendo as categorias e as páginas de conteúdo com um menu *popup*

Outra sugestão, que já havia sido apresentada na avaliação heurística, envolveu não deixar que os conteúdos criados ficassem automaticamente disponíveis no *site*, mas que o administrador pudesse visualizá-los antes de publicá-los.

Também foi sugerida a possibilidade de criação de subcategorias no menu vertical.

4.2.3 Conclusão sobre a avaliação piloto de uso

A avaliação piloto identificou alguns problemas que já haviam sido identificados anteriormente pela avaliação heurística como a necessidade de uma pré-visualização do conteúdo antes de sua publicação. Entretanto a maioria dos pontos levantados na avaliação piloto foram diferentes.

Muitos problemas só ficam evidentes quando o sistema já possui uma grande quantidade de dados ou quando ele é usado efetivamente. Como exemplo podemos citar um problema apontado no assistente de publicações, que se estende também para os demais assistentes: depois de serem ingressadas várias publicações no *site*, para alterar a ordem de uma publicação na lista é preciso clicar várias vezes pois só existem as opções de movê-la uma posição acima ou uma abaixo.

4.3 Considerações finais

A Avaliação Heurística e o teste de uso mostraram-se processos complementares no que tange a identificação de problemas na interface do software. O teste de uso revelou problemas que dificilmente seriam identificados na Avaliação Heurística, mesmo que esta tivesse sido executada por avaliadores experientes. Por outro lado, o fato da Avaliação Heurística ter sido executada antes da avaliação piloto de uso permitiu uma melhor qualidade no processo de desenvolvimento pois possibilitou a prevenção de uma série de problemas que dificultariam a realização do teste, pois muitos problemas identificados durante a Avaliação Heurística puderam ser corrigidos antes da avaliação piloto.

Capítulo 5

Conclusões

Nos últimos anos, a rede mundial de computadores - Internet – tem-se mostrado um meio eficiente para armazenamento e divulgação de informação. Esta característica foi rapidamente percebida pelos educadores que buscaram explorar este potencial. No entanto, a dificuldade de elaboração e manutenção de *sites*, inviabilizou inúmeras iniciativas. Esta dificuldade reside na obrigatoriedade de conhecimentos técnicos para criação de *websites*, quer sejam eles para uso de ferramentas para construção de páginas ou uso de linguagens técnicas como HTML. Além disso, a publicação dos *sites* também não é tarefa simples, já que necessita de transferência dos arquivos para máquinas com servidores web.

Além dos obstáculos encontrados para criação dos *sites*, a manutenção dos mesmos apresenta igual ou maior dificuldade. Este foi um dos problemas encontrados pelo grupo de EAD da Unicamp. O *site* da equipe foi desenvolvido com páginas estáticas e era mantido por várias pessoas, o que dificultava a tarefa de manutenção. Além disso, uma nova demanda para elaborar, manter e hospedar *sites* para projetos de professores e grupos da Unicamp ligados à EAD, motivou a busca por uma solução alternativa à criação estática de *websites*.

Para propor uma solução para resolver estes problemas, buscou-se investigar um conceito novo na criação de *sites*, baseado no gerenciamento de conteúdo. Vários sistemas gerenciadores de conteúdo (CMS) foram estudados dentre eles o PHP-Nuke e o Plone.

Essas ferramentas para criação e manutenção de *sites* na web apresentam uma série de problemas que impedem, ou no mínimo dificultam, seu uso por pessoas sem conhecimento profundo em informática. Tanto o PHP-Nuke quanto o Plone visam resolver o problema dos administradores de *sites*. Eles não tem por objetivo atingir o público sem conhecimentos técnicos em informática. O PHP-Nuke, por exemplo, tem uma estrutura

modular o que facilita a incorporação de novas funcionalidades. Entretanto, a criação dos módulos só pode ser feita por especialistas em informática com conhecimentos em PHP e que conheçam a estrutura da ferramenta. A criação de novas páginas ou tópicos depende da intervenção do administrador do *site* (especialista em informática) para alteração de módulos já existentes ou criação de novos. No entanto, esse sistema é um dos mais populares atualmente pela sua facilidade de instalação e incorporação de conteúdo depois de ter sido configurado adequadamente por um especialista.

O Plone, por outro lado, possui recursos tão sofisticados que o tornam uma ferramenta mais apropriada para *sites* de empresas e organizações que possuem uma grande quantidade de conteúdo a ser publicado. Dada a sua sofisticação, instalá-lo, configurá-lo e até mesmo usá-lo não é trivial para especialistas, muito menos para usuários sem conhecimentos técnicos em informática.

O conjunto de CMSs disponíveis no mercado que foram estudados não atendiam às necessidades do público que buscou na equipe de EAD da Unicamp um apoio para a criação de seus *sites*. Estas pessoas, na sua maioria professores da Unicamp, não possuem conhecimentos técnicos suficientes na área de informática e nem mesmo tempo para se especializarem na construção de *sites*. O objetivo destes professores é disponibilizar de forma rápida, fácil e eficiente o material didático utilizado em sala de aula para acesso pelos alunos, como uma forma de apoio ao ensino presencial.

Desta forma, como resultado desta pesquisa foi proposto um novo CMS, já apresentado no capítulo 3, denominado GECON. Seu objetivo é atender exatamente as pessoas com esse perfil trazendo a elas a possibilidade de criar e manter seus próprios *sites* sem a dependência de terceiros.

Para atingir esse objetivo, foram realizados testes de usabilidade ao longo do desenvolvimento do GECON, como descrito no capítulo 4. A característica básica desses testes é detectar problemas de usabilidade na interface relativos à facilidade de uso e de aprendizagem.

O *design* da interface do GECON foi feito de forma a promover a sua usabilidade. Além disso, o GECON possibilita a geração de *sites* com interfaces com boa estrutura de *design*. Procurou-se construí-lo de tal forma que os *sites* gerados tivessem um padrão pré-definido de disposição na tela dos elementos básicos de um *site*. Isto porque a falta de conhecimentos técnicos suficientes dos usuários do GECON poderia levar à criação de *sites* com pouca usabilidade.

Sua estrutura hierárquica permite que a correção de erros ou o aperfeiçoamento de funcionalidades seja feita uma única vez, independentemente do número de *sites* hospedados. Com isso, reduz-se sensivelmente o trabalho de manutenção da ferramenta.

Uma versão do GECON encontra-se instalada no servidor de EAD da Unicamp, onde é possível hospedar *sites* de projetos ligados à EAD. Durante o desenvolvimento do GECON optou-se por transformá-lo em software livre sob licença GPL (*General Public License*). Com isso, o sistema, e a sua documentação de instalação estão disponíveis para *download* no repositório de software livre da Unicamp.

Como extensões deste trabalho, vislumbra-se uma série de possibilidades. Inicialmente seria importante analisar e incorporar melhorias e aperfeiçoamentos no sistema, observados durante o uso efetivo do GECON no gerenciamento do *site* de EAD da Unicamp.

Para *sites* com mais de uma pessoa responsável pela atualização do conteúdo seria interessante limitar o acesso a partes do *site*, criando uma opção de bloqueio de edição por outros usuários, para garantir a autoria daquele conteúdo.

Depois de definido o design do *site*, a criação de conteúdo na área principal é a atividade mais freqüente. Talvez fosse adequado verificar se desvinculá-la das outras atividades, poderia facilitar a interação com a interface.

Outro trabalho futuro relacionado à ferramenta e que requer alguma alteração é criar *sites* segmentados. Por exemplo, uma empresa pode criar uma intranet com uma identidade visual única para todos os departamentos. Em cada um haveria um ou mais responsáveis pela atualização de sua sessão dentro do *site* da empresa, sem replicar a estrutura do GECON.

O sistema deveria ser testado com um número maior de pessoas com pouco conhecimento técnico em informática para averiguar qual porcentagem desse grupo efetivamente consegue publicar conteúdo na web quando utiliza de forma independente o GECON. Na medida em que pessoas leigas consigam utilizar o sistema eficientemente, poder-se-ia contribuir para a inclusão digital desse grupo de usuários.

Apesar de ter sido uma preocupação inicial no design da ferramenta, muito esforço é necessário no que se refere às questões relativas a acessibilidade. Também seria interessante investigar se o GECON promove a colaboração na criação dos *sites*

De forma geral, conclui-se que o desenvolvimento do GECON foi bastante positivo, dado que seu uso na reformulação do *site* de EAD da Unicamp mostrou-se satisfatório, facilitando a atualização do conteúdo e gerenciamento do *site*. Além disso, vários tutoriais

sobre o GECON têm sido proferidos no último ano, com um número significativo de pessoas interessadas na adoção do sistema.

Referências

ACCESS. Disponível em: <<http://office.microsoft.com/access>>. Acesso em: fev. 2003.

APACHE. Disponível em: <<http://www.apache.org>>. Acesso em: fev. 2003.

BRODER, T. Conheça as melhores tecnologias open source para criar sites dinâmicos. **Geek**, v. 4, n.31, p.20-23, 2003.

COCOON. Disponível em: <<http://cocoon.apache.org>>. Acesso em: fev. 2003.

DB2. Disponível em: <<http://www-306.ibm.com/software/data/db2>>. Acesso em: fev. 2003.

DIX, A.; FINLAY, J.; ABOWD, G.; BEALE, R. *Human-computer interaction*. New York: Prentice-Hall, 1993. 570p.

ISHII, H. (1993) Cross-cultural communication and CSCW. In: Harasim, L.M. *Global networks: computers and international communication*. Cambridge: MIT Press. p.143-151.

JEFFRIES, R., MILLER, J. R., WHARTON, C., UYEDA, K. M. User interface evaluation in the real world: A comparison of four techniques. **Proceedings of CHI 91**. New York: ACM, 1991. p. 119-124.

LAUREL, B.K. *The Art of Human-Computer Interface Design*, M.A.: Addison - Wesley, 1990

LENYA. Disponível em: <<http://cocoon.apache.org/lenya>>. Acesso em: fev. 2003.

LIMI, A.; RUNYAN, A. ANDERSEN, V. **The plone book**. 2000 Disponível em: <<http://www.plone.org/documentation/book>>. Acesso em: nov. 2003.

MACROMIDIA. Dreamweaver manual. Disponível em <<http://www.macromidia.com>> Acesso em: dez 2003.

MANDRAKE. Disponível em <<http://www.mandrake.org>> Acesso em: fev 2003.

MICROSOFT. Frontpage manual. Disponível em <<http://www.microsoft.com>> Acesso em dez 2003.

Referências

- MS-SQL. Disponível em: <<http://www.microsoft.com/sql>>. Acesso em: fev. 2003.
- MYSQL. Disponível em: <<http://www.mysql.org>>. Acesso em: fev. 2003.
- NARDI, B. *A small matter of programming: perspectives on end user computing*. Cambridge: MIT Press, 1993.
- NIELSEN, J. Homepage Usability: 50 Websites Deconstructed 2001
- NIELSEN, J. Designing Web Usability: The Practice of Simplicity. 2000
- NIELSEN, J. *Usability engineering*. Boston: Academic Press, 1993.
- NIELSEN, J.; MOLICH, R. Heuristic evaluation of user interfaces. In: EMPOWERING PEOPLE - CHI'90 CONFERENCE *Proceedings*. New York: ACM Press, 1990.
- NORMAN, D.; DRAPER, S. *User centered system design*. Hilldale: Lawrence Erlbaum Associates, 1986.
- OPENCMS. Disponível em: <www.opencms.org>. Acesso em: ago. 2003.
- ORACLE. Disponível em: <<http://www.oracle.com>>. Acesso em: fev. 2003.
- PHP-NUKE. **Advanced content management system**. Disponível em: <<http://www.phpnuke.org/>>. Acesso em: jul. 2003.
- PLONE. Disponível em: <<http://www.plone.org>>. Acesso em: fev. 2003.
- POKORNY, J. Static pages are dead: how a modular approach is changing interaction design. **Interactions**, v.8, n.5, p.19-24, 2001.
- POSTGRESQL. Disponível em: <<http://www.postgresql.org>>. Acesso em: fev. 2003.
- POSTNUKE. Disponível em: <<http://www.postnuke.com>>. Acesso em: jul 2003.
- ROBERTSON, J. **Choosing the right CMS authoring tools**. Sydney: Step Two Designs, 2003a. Disponível em: <http://www.steptwo.com.au/papers/kmc_authoringtools/>. Acesso em: out. 2003.
- ROBERTSON, J. **So, what is a content management system?**. Sydney: Step Two Designs, 2003b. Disponível em: <http://www.steptwo.com.au/papers/kmc_what/>. Acesso em: out. 2003.
- ROCHA, H. V. da; BARANAUSKAS, M.C.C. **Design e avaliação de interfaces humano-computador**. São Paulo: IME-USP, 2000. 242p.

Referências

SCAPIN, D. L. Organizing human factors knowledge for the evaluation and design of interfaces. **International Journal of Human-Computer Interaction**, 2, 1990. p. 203-229.

SQUISHDOT. Disponível em: <<http://www.squishdot.org>>. Acesso em: fev. 2003.

WEBMAKE. Disponível em: <<http://webmake.taint.org>>. Acesso em: fev. 2003.

WINOGRAD, T.; BENNET, J.; DEYOUNG, J.; HARTFIELD, B.,ed. *Bringing design to software*. New York: ACM Press; Reading: Addison-Wesley, 1996. 321p.

WYONA. Disponível em: <<http://www.wyona.com>>. Acesso em: fev. 2003.

XOOPS. Disponível em: <<http://www.xoops.org>>. Acesso em: fev. 2003.

ZOPE. Disponível em: <<http://www.zope.org>>. Acesso em: ago. 2003.

Referências

Anexo

Avaliações Heurísticas apresentadas.

Avaliação Heurística Grupo 1 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informã o usuário sobre o que está ocorrendo?</i>	<p>- O sistema informa o que está acontecendo, entretanto poderia ser mais explicativo, por exemplo, quando for feita alguma alteração, deveria ser dada uma mensagem mais explicativa sobre o que foi alterado, por exemplo, a mensagem que aparece se alterar o logo, banner é "Alterações Efetuadas", sendo que seria melhor se a mensagem fosse "O Logo foi Alterado com Sucesso" ou "O Banner foi Alterado com Sucesso";</p>	1
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	<p>- A maioria dos termos utilizados, são termos do mundo real, mas há alguns termos técnicos, por exemplo, pixels, banner e h1. Estes termos são comuns para quem trabalha com webdesign, porém são estranhos para a maioria da pessoas, dificultando assim o entendimento de algumas funções do sistema;</p> <p>- Sugerimos o uso de imagens associadas as funções para facilitar a utilização do sistema.</p>	2

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>3. Controle e Liberdade</p> <p>Ex. Há saída de emergência? Undo e Redo?</p>	<ul style="list-style-type: none"> - Não existe opção para desfazer uma operação efetuada, função indispensável para este tipo de sistema, aonde o usuário precisa testar varias idéias e escolher a que melhor se encaixa; - O sistema não permite o usuário criar novos objetos/caminhos. - Na página de Ajuda não tem como voltar para a página principal (menu); - Em todas as janelas que tem botão cancelar, deveria voltar para a página principal ou para página em que estava mexendo e não fechar a janela; - Quando aparece alguma página de erro default do Zope, o usuário não tem como voltar para a página em que estava, tendo que fechar o sistema e abrir de novo para continuar executando sua tarefas. 	5
<p>4. Consistência e Padronização</p> <p>Ex. Segue convenções de Plataforma?</p>	<ul style="list-style-type: none"> - As páginas são padronizadas utilizando botões e opções de escolha lembrando o padrão do Windows; - Em todas as páginas, os botões Cancelar e Confirmar estão fora do padrão normal, induzindo o usuário a cancelar a operação. A posição mais agradável seria o botão Confirmar à esquerda do botão Cancelar assim passamos a impressão que o usuário está utilizando o sistema corretamente. 	2

Heurística	Problemas Detectados	Grau [1-5] de Severidade
5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?	- Não previne a ocorrência de erros, por exemplo, foi testado uma outra extensão para figura (.PSD) e foi dada mensagem que a alteração foi efetuada, mesmo com o erro, pois os formatos possíveis são .GIF, .JPG e .PNG.	5
6.Reconhecer é melhor que relembrar Ex. É necessário lembrar informação de uma parte em outra?	- Deveria explorar mais o conceito de frames para facilitar o usuário a se localizar dentro do sistema, por exemplo, o menu principal poderia ficar fixo e não rolando junto com a página; - Os submenus estão confusos, deveria ser dado um destaque para a opção ativada. - Dentro de Opções Gerais do Site, não se tem clareza em que opção do sistema está (Elementos do Site, Página Principal, Mudar Aparência), seria interessante dar destaque.	2

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>7.Flexibilidade e Eficiência de Uso</p> <p><i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i></p>	<ul style="list-style-type: none"> - Os assistentes são bem limitados. - Não há flexibilidade, por exemplo, os botões Cancelar tem a função de fechar a janela, sendo que o mais prático seria ter a função de retornar para a página anterior; - O uso de Menu em todas as páginas facilita a vida do novato, mas o menu deveria ser fixo, pois rolando junto com a página, terá uma hora em que irá ficar escondido, assim sendo provavelmente o usuário irá esquecer do menu e fechar a janela, tendo que abrir novamente o sistema para fazer alterações. - O sistema é difícil e pouco intuitivo para o uso. O usuário deve navegar por todo o site para tentar descobrir onde alterar as opções desejadas 	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>8.Estética e Design</p> <p>Suficiente</p> <p><i>Ex. Contém informação irrelevante?</i></p>	<p>- O conteúdo é razoável e as informações são relevantes, mas precisa melhorar a estética e aumentar o número de informações sobre como utilizar o sistema, afim de tornar a interface com o usuário mais agradável e convidativa;</p> <p>- Sugerimos as seguintes correções:</p> <ul style="list-style-type: none"> - Nos links Banner, Parcerias e Rodapé, deveria ter como formatar os textos (tamanho, tipo, cor, negrito ou não) ; - No link Menu Vertical, deveria ter um combobox com todas as opções de fonte e deveria ter opções igual ao Menu Horizontal (Itens do Menu e Aparência do Menu); - No link Área Principal: <ul style="list-style-type: none"> - opção Assistente para páginas html simples : na visualização da página criada não deveria ter Fechar Janela e sim Retornar, porque a pessoa pode querer modificar algo que não gostou; - opção Assistente para Links : botão cancela deveria voltar para Incluir Assistente e não fechar a Janela. 	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>9. Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas?</i> <i>Sugere soluções?</i></p>	<ul style="list-style-type: none"> - Não indica qual é o problema específico. - Mostra uma página de erro default do Zope, e depois que mostra essa página, o usuário não tem como voltar para a página em que estava. - Concluindo, o sistema deveria ter um procedimento próprio de tratamento de erro (em português) com alternativas de como contornar o problema ou orientando a maneira correta de operar o sistema; - É necessário implementar um mecanismo para feedback de erros e encaminhamento de sugestões. 	5
<p>10. Ajuda e Documentação Há help/documentação com busca simples por tarefa?</p>	<ul style="list-style-type: none"> - Não existe ajuda e/ou documentação, infelizmente o usuário aprende a utilizar o sistema pela tentativa e erro; - O arquivo de help está sem conteúdo; - Não há busca simples por tarefa. 	5

Avaliação Heurística Grupo 1 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Muito precário a exibição de mensagens para o usuário final.	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Necessita de experiência na área de informática para utilizar o sistema.	3
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não possui.	4
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	Existem ações não identificadas assim como padronização na apresentação para o usuário final.	4
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	Não possui validação das informações preenchidas pelo usuário, assim como o retorno dos erros ocorridos.	4
6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	A existência deste problema não é percebida pelos usuários.	2

Heurística	Problemas Detectados	Grau [1-5] de Severidade
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	Não apresenta flexibilidade de utilização para os usuários novatos e experientes.	3
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Estética e design pobres que dificultam a utilização do sistema.	4
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	Não há indicadores de erros.	5
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Não possui help específico além de ser em inglês para um sistema todo em português.	5

Avaliação Heurística Grupo 2 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	O usuário não tem nenhuma informação sobre o que está acontecendo. O sistema não possui mensagens para orientação.	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Utiliza palavras do mundo real, direcionadas para a proposta do Sistema, mas ainda existem problemas, como o usuário precisar saber o código das cores.	2
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não existem teclas de Undo e Redo, ou seja, tem que se dar continuidade ou cancelar a ação, necessitando assim reiniciar o processo.	4
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	Padrão para as teclas não são seguidos , como por exemplo a Teclas de Fechar as Janelas não segue o Padrão das Teclas de Confirmar e Cancelar.	2
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	O Sistema não se comunica com o usuário com relação a erros, não existe qualquer tipo de prevenção neste sentido.	4
6.Reconhecer é melhor que lembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	As funções que são utilizadas ao longo do sistema deveriam estar em todas as telas, facilitado assim a navegação, evitando que o usuário tenha que voltar telas para executar uma ação.	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	Não utiliza recursos de flexibilidade de uso, ou seja, o mais experiente tem que seguir o mesmo procedimento se um novato.	3
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Não existem informações, o design é confuso de difícil compreensão para o usuário leigo, botões não são auto explicativos	4
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	Sem indicação de erros e muito menos como resolver estes erros.	5
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Existe link de ajuda, mas não há informação concreta sobre a página. O link Geral indica somente os elementos de alteração da página, também sem conteúdo como descrição de alterações.	5

Avaliação Heurísticas Grupo 2 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Apresenta pouco feedback, dificultando o entendimento do usuário a respeito do funcionamento dos ícones e menus.	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Em determinados menus apresenta termos técnicos, sem que haja a devida explicação sobre os seus significados.	2
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não apresenta opções de Undo e Redo .	4
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	A tela apresenta-se confusa não informando atalhos para usuários leigos.	3
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	Não, efetivamente não apresenta prevenção de erros.	5
6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	Apresenta padrões entre as telas facilitando a navegação do usuário.	1

Heurística	Problemas Detectados	Grau [1-5] de Severidade
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	Nas telas de função de construção do site encontramos umas funções de atalhos. Mas achamos interessante o surgimento de outras.	2
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Apresenta um design confuso para um usuário leigo, não informando como utilizar as funcionalidades. Mas, não encontrei itens irrelevantes.	3
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	Não encontramos informes de erros.	5
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Não informa nenhum help consistente para explicar o funcionamento da página.	4

Avaliação Heurística Grupo 3 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Informa, mas não de forma muito clara. Difícil iniciar a construção da página. O usuário precisa ter noções de uso da web.	3
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Utilizou-se um ícone muito pequeno e distorcido para a edição do local. As dificuldades na compreensão de termos técnicos podem aparecer. Algumas palavras difíceis para o usuário leigo, principalmente na parte da área principal.	3
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não há controle real do usuário sobre o sistema. Não existe a possibilidade de desfazer a alteração e nem ajustar a imagem. O usuário fica limitado quanto à formatação da janela.	5
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	Sim, está padronizado.	1
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	O usuário leigo encontra dificuldades com as mensagens, pois não dizem onde corrigir os problemas.	3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	Um mapa do site melhoraria e muito Ou existir um botão de visualização da página.	2
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	O usuário avançado e o novo tem os mesmos recursos. A rotina para o usuário experto é cansativa.	4
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Existem sim informações irrelevantes. A estética é muito ruim A visualização do site é muito demorado. Os ícones muito pequenos. Site sem informações iniciais de montagem.	5
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	As mensagens não são claras e não indica soluções. Não avisa o tipo de erro e uma ajuda de como corrigi-lo.	3
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	A ajuda é bem simples e detalhada. Mas usuários iniciantes podem sentir dificuldades.	3

Avaliação Heurística Grupo 3 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Não, e o mínimo de informações que o sistema possui é confusa, e os problemas não estão destacados, e quando aparece algum tipo de problema, vem munido de códigos dificultando um usuário leigo no assunto.	5
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Sim contem informações do mundo real, como as imagens, por exemplo, mais como esta pagina esta em constante mutação, encontramos termos técnicos, na segunda semana.	4
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não estão disponíveis os programas UNDO e REDO, mas há sistemas de cancelamento caso haja algo errado.	2
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	Em alguns, aspectos encontramos ícones com acesso diferenciado. Isto é de uma semana para outra o usuário comum ira confundir-se, pois o sistema Gecon não segue um padrão.	5
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	Ao testar problemas na paginas, não obtive respostas negativas, o programa continuou a rodar sem problemas.	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	Não, todas as informações estão situadas na tela, e ao salvar outro item, basta trocar a opção.	2
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	O sistema utiliza atalhos simples e de fácil entendimento, sendo escasso em modos mais avançados.	2
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Mau designer tanto para visualização integral do sistema, quanto à contextualização, nota-se a ausência de estética.	5
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	Mesmo errando, o sistema não envia mensagens para indicar o erro. Seria necessárias que detectasse o erro das informações, antes que fosse cometido, ocasionando assim maior produtividade.	5
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Embora haja link de ajuda, não estavam disponíveis neste site.	5

Avaliação Heurística Grupo 4 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>1. Visibilidade</p> <p><i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i></p>	<p>O sistema não possui nome, no topo das telas, o que já ajudaria a identificar a função.</p> <p>O sistema não informa ao usuário o que está acontecendo. A primeira página, por exemplo, não tem como saber qual a finalidade do site. As outras páginas não informam, por exemplo, o que é um banner, pra que serve, onde fica. Então a pessoa pode configurar um banner sem saber o que vai acontecer no site.</p> <p>O botão “Cancelar” de todos os menus, sai da tela onde o usuário está trabalhando e volta para a tela inicial, sem qualquer aviso. O sistema permite alteração em qualquer parte do site, esteja esta parte visível ou invisível, sem avisar ao usuário. Na confirmação da criação de um “Assistente de Link” da área principal, o sistema volta para tela de criação de link, sem dizer se o link foi criado ou não.</p>	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i></p>	<p>Há muitos termos técnicos. Ex: “Coloque na área abaixo um texto simples ou um texto com marcações html:” . Browse – utilizada para buscar um arquivo. Todas as cores são identificadas por código. Tamanho de área de logotipo identificado em pixels. Tamanho de fonte também identificado por código. As próprias palavras Logo e Banner. Assistente de eventos não diz nada para um usuário novato.</p>	4
<p>3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i></p>	<p>O botão “cancelar” deveria voltar uma alteração feita em uma determinada parte do site, porém retorna para a tela inicial, fazendo o mesmo efeito do “fechar janela”. Os botões Undo e Redo não foram identificados no sistema. Temos que refazer novamente, não há como simplesmente desfazer ou refazer uma ação.</p>	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>4.Consistência e Padronização Ex. Segue convenções de Plataforma?</p>	<p>Não possui padronização de caixas, botões e telas de um modo geral. Por exemplo, está em forma de botão o cancela e confirma, mas fechar janela não está, assim como as subopções das janelas (Incluir Assistente, Excluir assistente, etc.)</p> <p>Não há posicionamento do cursor.</p> <p>“Código de cor”, ao invés da própria cor escolhida na paleta de cores. “Código do fonte”, ao invés da lista de escolha de tamanhos pré-determinados. “Nome do fonte” sem a lista de opções para escolha.</p> <p>O símbolo do ícone “Editar”, na tela inicial, lembra o MS Paint e o usuário imagina que vai sair desenhando e se depara com menus (frustração).</p> <p>O próprio sistema não mantém um padrão definido, por exemplo, nos Assistentes da Área Principal cada confirmação de um tipo de assistente aparece de um jeito diferente.</p> <p>Faltam os ícones novo, abrir, salvar, imprimir, visualizar, recortar, copiar e colar no menu principal.</p>	<p>5</p>

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?</p>	<p>Quando o usuário entra no sistema, ele tenta realizar alguns testes nas opções dos menus já criados e aparecem mensagens de erros. Ao clicar nos vários ícones “Editar”, que não possuem identificação clara, ele se depara com diferentes menus e busca “Ajuda”, não consegue nada. Se ele for muito persistente ele tenta navegar pelo “Geral” e aos poucos vai entendendo o que ocorre no sistema, caso contrário, não volta.</p> <p>Não há sinais sonoros quando ocorre um erro. Por exemplo, se o Menu Vertical ficou marcado para ser invisível, ao escolher esta opção, o sistema deveria emitir um sinal sonoro de erro. Para que não ocorra esse erro, o ideal seria escolher os elementos do site, e nas próximas páginas, estivessem disponíveis apenas os elementos escolhidos para constar no site.</p> <p>O sistema não solicita confirmação (dupla) de ações que podem gerar perdas de dados e/ou resultados. No caso de excluir um parceiro por exemplo, o sistema não pede confirmação.</p>	<p>5</p>

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i></p>	<p>Está sendo necessário lembrar uma informação de uma parte em outra. Exemplos: Se o usuário não lembrar que deixou o “banner” invisível no menu “Geral”, ele altera todo o “banner” e fica aguardando a sua visualização no site. Outro exemplo é mensagem de confirmação do “Assistente de Publicações”: A nova categoria “ “ foi criada com sucesso (“ “ em branco).</p>	4
<p>7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i></p>	<p>Os ícones “Editar” da tela inicial talvez tenham sido uma tentativa de criação de aceleradores visíveis, porém causam confusões nos novatos e até mesmo em “expertos”. Já os aceleradores invisíveis como teclas de atalho, como exemplo Ctrl+C ou Ctrl+V, não existem.</p>	3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i></p>	<p>O sistema contém informações irrelevantes e ao mesmo tempo omite informações importantes.</p> <p>O design, por exemplo, deve ser suficiente e não omissivo. Exemplo: os ícones “Editar” em grande quantidade, por si só não dizem o que significam. O menu principal deveria ter mais opções e direcionar os trabalhos do usuário.</p> <p>Já na caixa de cores, não há necessidade de mostrar o código das cores. Seria ideal que mostrasse a própria cor dentro da caixa e mais nada.</p>	4
<p>9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i></p>	<p>Não indica problemas, nem soluções</p> <p>Nas mensagens de erro temos uma identificação técnica do erro com sugestões mais técnicas ainda.</p> <p>Complementando, o erro e as possíveis soluções são apresentadas em inglês.</p>	4
<p>10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?</p>	<p>Não há ajuda No botão “Ajuda”, temos um link para o site Zope. No site Zope tudo está em inglês, não se parece nada com a ferramenta que o usuário está utilizando e não possibilita nenhuma busca pelo tópico que precisa ser esclarecido. Também possui uma data de atualização de 2001.</p>	5

Avaliação Heurística Grupo 4 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	A primeira pagina é muito confusa, não consegui definir tamanhos para as imagens.	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Há algumas palavras que usuários leigos poderão não saber o significado, como os do menu da área principal, ou seja, os assistentes de HTML, de eventos, de links e de publicações. Por exemplo: se quiser saber para que serve o assistente de HTML é preciso escolher primeiro, na opção incluir um assistente, o assistente de HTML para saber o que é. Usuário poderá ficar meio perdido ou com “preguiça” de ficar clicando daqui e dali para entender o que são cada um destes assistentes.	4
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não há saídas de emergencia	5
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	Sim, o sistema busca um determinado padrão nas telas do gerenciador principal do sistema, esse padrão deverá ser melhorado.	3
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	Não, um exemplo é o tamanho da imagem do logo ele não aceita a formatação que você sugere, há falhas por todo lado	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
6.Reconhecer é melhor que lembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	É difícil neste caso reconhecer.	5
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	Não, o sistema possui uma interface única	5
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Está meio confuso	4
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	<p>Não há nada sobre erros. Por exemplo, se o usuário escolher uma imagem muito grande que não cabe no espaço especificado, não há nenhuma mensagem que informa sobre isso.</p> <p>Tem que ficar clicando no botão confirma para ver na tela se tudo está ok. Mas as vezes não aparece nada na tela; deduz-se então que não deu certo a escolha da imagem.</p>	5
10.Ajuda e Documentação <i>Há help/documentação com busca simples por tarefa?</i>	Não esta funcionando	5

Avaliação Heurística Grupo 5 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Site confuso para os usuários e com poucas mensagens de retorno, ou seja, falta feedback.	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Algumas palavras devem ser substituídas para uma melhor compreensão do usuário, como por exemplo pixels, importar. O ícone da "latinha com o lápis" deveria ser substituído para uma caneta nos casos onde não se pode carregar figuras.	3
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não existe saída de emergência e também não existem as funções UNDO e REDO.	4
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	Não faz consistência das informações fornecidas pelo usuário, em determinadas partes do sistema, se o usuário não fornecer a informação pedida e confirmar, o sistema emite uma mensagem de "Alterações Efetuadas". Os links não seguem o padrão Internet, como sublinhados.	3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?</p>	<p>Não prevê erros temos 2 exemplos:</p> <p>1) Na opção Assistente de eventos ocorreu um erro totalmente inesperado e com uma mensagem totalmente desconhecida. Zope has encountered an error while publishing this resource. Error Type: AttributeError Error Value: novo_conteudo_html</p> <p>2) Caso o usuário escolha Área Principal e não selecione nenhum option box e clicar em confirma o sistema exibe uma mensagem Error Type: KeyErro; Error Value: assistente. Seria interessante que o sistema verifique se alguma opção foi selecionada e em caso negativo mostre uma mensagem "Marque alguma opção!!!</p>	5
<p>6.Reconhecer é melhor que lembrar Ex. É necessário lembrar informação de uma parte em outra?</p>	<p>Não existe um help do sistema e conseqüentemente não há como recuperar as instruções.</p>	4
<p>7.Flexibilidade e Eficiência de Uso Ex. Usa aceleradores (in) visíveis para o (novato) experto?</p>	<p>Existe uma barra de botões acima do site que ajuda o usuário a não ter que ficar toda hora fechando e o modo de administração, porém estes botões deveriam estar + visíveis, sem que o usuário tenha que utilizara a barra de rolagem vertical e horizontal</p>	3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	Após o acesso à opção Assistente para páginas html simples, solicitou-se a criação de um nova pasta e o feedback foi: (HTML 683821 incluída.). A informação 683821 esta totalmente fora do contexto	5
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	No caso do usuário confirmar sem fornecer as informações pedidas, a mensagem de erro é fornecida em inglês, além de ser uma mensagem técnica inacessível para um novato. Como exemplo podemos citar o acesso a opção Assistente de Eventos	5
10.Ajuda e Documentação <i>Há help/documentação com busca simples por tarefa?</i>	Não existe help, apenas algumas mensagens insuficientes para orientar o uso das opções	5

Avaliação Heurística Grupo 5 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	<p>As vezes parece que o usuário não percebe o componente que está sendo manipulado;</p> <p>Realmente até eu conseguir validar as alterações levou um tempo, pois não existe nada informando que eu necessito fechar a janela para que as minhas alterações apareçam na página principal.</p>	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	<p>O usuário precisa saber o código da cor no mapa de cores para alterar para a desejada;</p> <p>Não existe uma indicação mais facilitada dos componentes que integram uma página;</p> <p>Link quando ativado, menu, são coisas que pessoas leigas, podem até estar acostumadas a ouvir, mas não sabem o que são e nem qual a sua funcionalidade.</p>	3
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	<p>Não possui funções de undo/redo e não permite que o usuário teste as mudanças realizadas antes de efetiva-las;</p> <p>Não permite que o usuário salve o layout definido para uso posterior;</p>	3
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	<p>A tecla de fechar a janela não segue o padrão das de Cancelar / Confirmar, se parecendo mais como texto puro.</p>	2

Heurística	Problemas Detectados	Grau [1-5] de Severidade
5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?	Não impede que o usuário indique arquivos que na realidade não são imagens para serem colocadas no site; Não existe uma indicação clara dos limites da interação do usuário com o sistema; As etapas devem ser efetuadas de forma correta.	3
6.Reconhecer é melhor que relembrar Ex. É necessário lembrar informação de uma parte em outra?	Com relação a esse aspecto, não necessitamos relembrar nada de uma parte para outra;	1
7.Flexibilidade e Eficiência de Uso Ex. Usa aceleradores (in) visíveis para o (novato) experto?	Não existe mais opções da forma que o usuário pode configurar o site, por exemplo uma opção que permitisse a configuração em bloco dos componentes; Não oferece configurações já definidas que o usuário poderia utilizar como base; Muitíssimo Lento, stressante	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>8.Estética e Design</p> <p>Suficiente</p> <p><i>Ex. Contém informação irrelevante?</i></p>	<p>Os tamanhos das fontes na tela de administração do site deveriam ser organizadas de acordo com a sua importância e função;</p> <p>Possui uma interface que não está adequada aos padrões atuais da Internet</p> <p>Com relação aos botões existentes, por exemplo, no momento em que clico em geral, um usuário leigo, afinal esse é o propósito da página, não saberá que Elementos do Site, Página Principal, são botões que podem ser clicados, não dá esse enfoque, não é intuitivo como o botão confirma e cancela.</p> <p>Outro exemplo, mapa de cores, tem usuário que não vai saber que ali se deve clicar, daí tem a frase, escolha uma cor, mas onde ???</p>	4
<p>9.Auxilia no reconhecimento, diagnóstico e recuperação de erros?</p> <p><i>Ex. Indica problemas? Sugere soluções?</i></p>	<p>Não possui funções onde o usuário possa ser “encaminhado” a realizar ou recuperar algo que tenha feito anteriormente, como os famosos “Wizards”;</p> <p>Não existe nenhuma ferramenta com relação a isso, HINTS, ajudas.</p>	3
<p>10.Ajuda e Documentação</p> <p>Há help/documentação com busca simples por tarefa?</p>	<p>Não proveem nenhuma ajuda, mesmo a usuários novatos;</p> <p>Não permite que usuários experientes possam aproveitar melhor a ferramenta por não prover nenhuma documentação.</p>	5

Avaliação Heurística Grupo 6 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>1.Visibilidade</p> <p><i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i></p>	<p>Concordamos quanto às mensagens de feedback, que informam sobre as alterações feitas pelos usuários após a sua confirmação e que são enviadas com rapidez.</p> <p>Sentimos a falta de algumas informações como o tooltip para auxiliar na identificação dos botões. Não sabemos quando inserimos um logotipo ou um menu na página principal.</p>	2
<p>2.Coerência</p> <p>Sistema/Mundo Real</p> <p><i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i></p>	<p>Sim, estão coerentes com o mundo real.</p>	1
<p>3.Controle e Liberdade</p> <p><i>Ex. Há saída de emergência? Undo e Redo?</i></p>	<p>Há a flexibilidade de fechar as janelas a qualquer momento. Porém, não existe funções de undo e redo, tirando a liberdade total e controle do usuário, já que eles podem escolher funções por engano e não terão como voltar às funções anteriores.</p>	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>4.Consistência e Padronização Ex. Segue convenções de Plataforma?</p>	<p>Existe a padronização de alguns labels, mas algumas entradas como o tipo de fonte estão fora do padrão. O usuário teria que adivinhar qual o tipo de fonte usaria, pois não há menu como nas outras opções.</p> <p>As barras normalmente ficam à esquerda e neste caso os menus estão na parte superior e lateral direita.</p> <p>Não há consistência nos campos de entrada das configurações (cores, fontes, etc). O usuário pode entrar com qualquer valor.</p> <p>Quanto às palavras chave utilizadas nos links estão corretas, pois seguem as convenções da plataforma computacional.</p>	2
<p>5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?</p>	<p>Inserimos alguns arquivos que não existiam ou que eram incompatíveis, mas o sistema simplesmente aceitou e não retornou nenhuma mensagem. Apesar do sistema não cair, é um erro que deixa o usuário perdido, pois ele provavelmente não saberá o que está acontecendo no seu sistema.</p> <p>Não há a prevenção deste tipo de erro e o envio de mensagens de aviso ao usuário.</p>	4

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i></p>	<p>Existem informações nas telas de administração suficientes para a maioria dos casos. Porém, sentimos a falta de instruções quanto à identificação dos ícones de inserção das figuras. Não há nenhuma informação sobre eles a não ser o tooltip “Editar”.</p>	3
<p>7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i></p>	<p>O sistema contém atalhos, mas não aceleradores para o novato esperto.</p>	3
<p>8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i></p>	<p>Faltam algumas informações na página inicial, mas não há informações irrelevantes.</p>	2
<p>9.Auxília no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i></p>	<p>Provocamos alguns erros, mas não recebemos mensagens de retorno. Os erros de atributos no Zope mostram a tela de erro de sistema e não uma mensagem de erro prevista. Ou seja, o sistema não ajuda na identificação, diagnóstico e correção dos erros.</p>	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Há vários links para ajuda, embora eles caiam em uma página do Zope e não em um documento desenvolvido especificamente para o sistema. Não encontramos o recurso de busca que também auxilia muito o usuário.	5

Avaliação Heurística Grupo 6 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	O sistema não tem um feedback para informar ao usuário o que está ocorrendo. Sugerimos adotar um mecanismo de feedback para melhorar.	4
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	As convenções são de fácil compreensão, bastando para melhorá-las ainda mais, apenas alguns detalhes cosméticos.	2
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Como foi apontado por todos, a falta de recursos como Undo e Redo comprometem a desenvoltura do usuário. É muito ruim termos um sistema que não podemos errar. A ausência desse simples recursos pode fazer o usuário perder tempo, se aborrecer e até mesmo desistir.	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
4.Consistência e Padronização Ex. Segue convenções de Plataforma?	Padronização é razoável, pois o usuário aprendendo uma operação irá tendo mais facilidade em seguir devido à padronização adotada	2
5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?	Não tem mecanismos de prevenção de erros e quando um erro é feito, sua mensagem na tela deixa o usuário leigo confuso e sem ter muito o que fazer, lembrando muito o Windows com aqueles erros que você não tem saída	4
6.Reconhecer é melhor que relembrar Ex. É necessário lembrar informação de uma parte em outra?	Como a padronização é boa, o usuário reconhece as funções e vai tendo mais desenvoltura.	2
7.Flexibilidade e Eficiência de Uso Ex. Usa aceleradores (in) visíveis para o (novato) experto?	Baixa flexibilidade e a eficiência é comprometida em função das falhas apresentadas acima.	4
8.Estética e Design Suficiente Ex. Contém informação irrelevante?	A estética e o design do site são ruins, cabendo aqui um amplo retrabalho para o projeto ter maior atratividade.	4
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? Ex. Indica problemas? Sugere soluções?	O sistema não possui mecanismos para auxiliar o usuário no reconhecimento dos erros e na recuperação.	3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
10. Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Na questão de ajuda e documentação, é mais referente à algumas operações, sendo um pouco limitado. Seus recursos de ajuda são muito básicos	3

Avaliação Heurística Grupo 7 A

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1. Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Inicialmente, não há informação da proposta do site	3
	Somente figuras, nenhuma explicação de uso	4
	Não. Na primeira vez fiquei meio perdida.	4
	Não, logo de cara na primeira página de acesso o usuário fica perdido, poucas informações	5
	O sistema não informa o usuário o que deve ser feito.	5
	Não informa nada. Se eu não soubesse que do se trata acharia que era uma engano!	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
<p>2.Coerência</p> <p>Sistema/Mundo Real</p> <p><i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i></p>	<p>ícones de pincéis para editar, mas sem explicação</p> <p>mapa de cores usa notação hexadecimal ao invés da cor</p> <p>pixels não é um termo comum ao usuário</p> <p>Não usa convenções do mundo real. Alguns termos complicados para iniciantes, como também convenções difíceis, como por exemplo ter que saber o tamanho ocupado pelo logo em pixels ou porcentagem</p> <p>As palavras e termos apresentados até que são simples, porém mesmo assim para pessoas que não tem um conhecimento básico torna-se complicado</p> <p>Não existem convenções do mundo real no sistema e o agravante é termos orientados ao sistema também estão mal apresentados</p> <p>Usa termos totalmente fora do mundo do usuário que quer começar uma página da WEB.</p>	<p>3</p> <p>3</p> <p>3</p> <p>5</p> <p>3</p> <p>5</p> <p>4</p>

Heurística	Problemas Detectados	Grau [1-5] de Severidade
3. Controle e Liberdade Ex. Há saída de emergência? Undo e Redo?	Não existe undo, redo Não existe meio para retroceder aquilo já feito. Não, nem de voltar, undo e redo. Todas alterações são on-line uma vez feita é realizada. Voltar apenas do Browse Não existe possibilidade de desfazer alguma alteração, para que isso ocorra deve-se realizar novamente a operação. Não possui correção	4 5 3 3 5
4. Consistência e Padronização Ex. Segue convenções de Plataforma?	Os links não possuem cor diferente do resto do texto Existe um padrão nos comandos de editar, após o usuário identificar se torna claro, o problema maior é identificar, pois esse comando é mal identificado Em alguns casos como por exemplo de botões (cancela, confirma, fechar janela, etc) mas mesmo assim possui botões sem explicação Existe um padrão nos comandos de editar, após o usuário identificar se torna claro, o problema maior é identificar, pois esse comando é mal identificado Não possui instrução clara de identificação	3 3 2 3 3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
5.Prevenção de Erros Ex. Previne a ocorrência de Problemas?	Não há prevenção de erros Previne. Não, o sistema informa que determinadas opções não podem ser acessadas em virtude de conter erros. O sistema previne erros. Não previne erros	3 1 4 1 3
6.Reconhecer é melhor que relembrar Ex. É necessário lembrar informação de uma parte em outra?	Ao iniciar o processo de definições de opções (Dependendo do browser em uso) , a janela que se abre não exibe todo seu conteúdo, necessitando maximizar É necessário lembrar algo entre duas fases Sim. Sim, já que o programa é feito para usuários leigos. Portanto é sempre bom lembrar Sim é necessário. Há necessidade de relembrar comandos.	2 2 3 1 3 3

Heurística	Problemas Detectados	Grau [1-5] de Severidade
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	<p>Não há como voltar a pagina anterior, somente ALT + left disponível pelo browser</p> <p>Para mudar de assunto (ex: de área principal/assistente qualquer/ após incluir item para geral) é necessário fechar janela e abrir de novo, pois causa erro</p> <p>O nível de flexibilidade existe mas é confuso.</p> <p>Sim, no caso botões sem significado que podem ser utilizados por usuários mais experientes.</p> <p>O nível de flexibilidade existe mas pode confundir até usuários mais experientes.</p> <p>Não existem atalhos</p>	<p>5</p> <p>5</p> <p>1</p> <p>1</p> <p>3</p> <p>3</p>
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	<p>Estética e design simples, minimalista e as vezes insuficiente</p> <p>Sim.</p> <p>Sim , para uma versão inicial é bem simples, menus rápidos, telas simples.</p> <p>Não.</p> <p>Falta informação</p>	<p>2</p> <p>2</p> <p>1</p> <p>1</p> <p>2</p>

Heurística	Problemas Detectados	Grau [1-5] de Severidade
9.Auxília no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas?</i> <i>Sugere soluções?</i>	Assistente de eventos causa erro sem explicação Dependendo do browser em uso, ocorre erro que impossibilita o uso Ao manipular um item (ex:área principal) e tentar mudar para outro, ocorre erro inexplicável Não. Sim, do tipo: Error Type: AttributeError Error Value: __call__. Porém deverá ser interpretado por algum usuário mais experiente. Não são apresentadas mensagens de erros. Não existem indicações.	5 5 5 3 3 1 3
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	há texto de Ajuda, mas de acesso difícil e muito tecnico Existe uma opção de ajuda, mas muito complicada Sim, porém de difícil entendimento. Existe uma opção de ajuda, mas esta é muito complicada para um usuário não avançado. Não carregou Ajuda	4 5 4 5 5

Avaliação Heurística Grupo 7 B

Heurística	Problemas Detectados	Grau [1-5] de Severidade
1.Visibilidade <i>Ex. O sistema informa o usuário sobre o que está ocorrendo?</i>	Não existe uma clara definição para o usuário do que exatamente está acontecendo. Os menus são realmente confusos e o usuário não consegue definir qual o objetivo do site.	5
2.Coerência Sistema/Mundo Real <i>Ex. Usa convenções do mundo real em vez de termos orientados ao sistema?</i>	Não existe ordem sistêmica nem analogia com o mundo real. É pouco usual e de difícil compreensão em um primeiro momento.	5
3.Controle e Liberdade <i>Ex. Há saída de emergência? Undo e Redo?</i>	Não existe uma rota definida para ir e voltar. O usuário se perde no meio do site não sabendo exatamente onde ele está. As alterações não podem ser desfeitas.	3
4.Consistência e Padronização <i>Ex. Segue convenções de Plataforma?</i>	O layout é confuso e fora dos padrões normalmente utilizados.	4
5.Prevenção de Erros <i>Ex. Previne a ocorrência de Problemas?</i>	Como o layout está mal elaborado, torna-se impossível a prevenção de erros, pois as ações do usuário não são previsíveis.	5

Heurística	Problemas Detectados	Grau [1-5] de Severidade
6.Reconhecer é melhor que relembrar <i>Ex. É necessário lembrar informação de uma parte em outra?</i>	O site tenta gerar intuitividade para o usuário. Depois de mexer um pouco no site, ele se tornou um pouco mais fácil de manipular. Mas falta informação ao usuário.	3
7.Flexibilidade e Eficiência de Uso <i>Ex. Usa aceleradores (in) visíveis para o (novato) experto?</i>	Existe uma tentativa de proporcionar ao usuário aceleradores, porém inócua. Talvez o usuário experto consiga navegar um pouco mais rápido após adquirir mais experiência no site.	3
8.Estética e Design Suficiente <i>Ex. Contém informação irrelevante?</i>	O sistema de informação ao usuário é deficiente. É difícil avaliar o conteúdo das informações pois elas são inexistentes.	3
9.Auxilia no reconhecimento, diagnóstico e recuperação de erros? <i>Ex. Indica problemas? Sugere soluções?</i>	As mensagens de erro são superficiais e não relevantes. Deveria ter um mapa do site, help, menus bem definidos, tips (dicas) e um template mais bem definido. O usuário não consegue perceber os erros cometidos, impedindo-o de sugerir melhorias além dele ficar perdido na construção do site.	3
10.Ajuda e Documentação Há help/documentação com busca simples por tarefa?	Existe a janela de help, apesar dela não ser sensível ao contexto. O help não possui um formato já conhecido pelo usuário além de ser pouco elucidativo. A Ajuda deve ser mais detalhada.	4