

Unioeste - Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Informática
Curso de Bacharelado em Informática

Usabilidade em Sistemas Web - Um Estudo de Caso

Anselmo Luiz Éden Battisti

CASCADEL
2007

ANSELMO LUIZ ÉDEN BATTISTI

Usabilidade em Sistemas Web - Um Estudo de Caso

Monografia apresentada como requisito parcial
para obtenção do grau de Bacharel em Informática,
do Centro de Ciências Exatas e Tecnológicas da
Universidade Estadual do Oeste do Paraná - Cam-
pus de Cascavel

Orientador: Prof. Carlos José Maria Olguín

CASCADEL
2007

ANSELMO LUIZ ÉDEN BATTISTI

Usabilidade em Sistemas Web - Um Estudo de Caso

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Informática, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel, aprovada pela Comissão formada pelos professores:

Prof. Carlos José Maria Olguin (Orientador)
Colegiado de Informática, UNIOESTE

Prof(a). Claudia Brandelero Rizzi
Colegiado de Informática, UNIOESTE

Prof. Ivonei Freitas da Silva
Colegiado de Informática, UNIOESTE

Cascavel, 22 de outubro de 2007

AGRADECIMENTOS

Aos meus familiares, amigos e professores que tanto colaboraram para que este trabalho fosse realizado.

Lista de Figuras

2.1	Estatísticas sobre o percentual de uso dos navegadores mais populares.	7
2.2	Representação de <i>select</i> nos navegadores: (a) Mozilla Firefox 2.0, (b) Internet Explorer 7.0 e (c) Opera 8.0.	8
2.3	Interface desenvolvida utilizando a biblioteca EXT	9
2.4	Painel de configuração de layout do Yahoo	12
2.5	Exemplo do DOM de uma página HTML	13
3.1	Machado manual proveniente da Espanha do período Paleolítico Inferior.	14
3.2	Parte do formulário de acadêmico do sistema SISO.	17
3.3	Mensagem de erro sobre falhas no preenchimento de campos do formulário.	18
3.4	Ambiente controlado para testes de usabilidade.	22
4.1	Módulos principais do sistema SISO.	27
4.2	Distribuição do uso dos navegadores.	28
4.3	Conhecimento sobre Linux.	29
4.4	Botões de ação do formulário de cadastro de acadêmicos.	31
4.5	Primeiro problema de usabilidade.	31
5.1	Formas de distribuir os botões em um formulário.	33
5.2	Formulário para cadastrar projetos no DotProject.	34
5.3	Parte do cadastro de aluno.	35
5.4	Ficha de inscrição para o vestibular 2008 da Unioeste.	35
5.5	Diagramas de formulários onde o uso de AJAX é aconselhado.	36
5.6	Formulário para solicitar paciente.	36
6.1	Fila de espera da clínica odontológica.	39

Lista de Tabelas

2.1	Arquitetura Cliente X Arquitetura Servidor.	6
3.1	Severidade do problema detectado.	21
3.2	Frequência com que os problema detectado ocorre.	21

Lista de Abreviaturas e Siglas

SISO	Sistema Odontológico
CO	Clínica Odontológica
SGBD	Sistema Gerenciador de Banco de Dados
HTML	<i>Hiper-Text Markup Language</i>
W3C	<i>World Wide Web Consortium</i>
CSS	<i>Cascading Style Sheet</i>
ISO	<i>International Organization for Standardization</i>
ABNT	Associação Brasileira de Normas e Técnicas
DR	Documento de Requisitos
SUS	Sistema Único de Saúde
CEO	Centro de Especialidades Odontológicas
NIT	Núcleo de Inovações Tecnológica

Sumário

Lista de Figuras	v
Lista de Tabelas	vi
Lista de Abreviaturas e Siglas	vii
Sumário	viii
Resumo	x
1 Introdução	1
1.1 Objetivos	1
1.2 Usabilidade	2
1.3 SISO - Sistema Odontológico	2
1.4 Organização do Trabalho	3
2 Sistemas Web	5
2.1 Arquitetura Cliente-Servidor	6
2.2 Navegador Web	7
2.3 Aplicações Ricas no Navegador Web	8
2.3.1 <i>Applet Java</i>	9
2.3.2 Flash	10
2.3.3 AJAX	11
3 Usabilidade	14
3.1 O Que é Usabilidade?	15
3.1.1 Facilidade de Aprendizado	16
3.1.2 Eficiência de Uso	17
3.1.3 Facilidade de Memorização	17
3.1.4 Baixa Taxa de Erros	18

3.1.5	Satisfação Subjetiva	18
3.2	Testes de Usabilidade	19
3.2.1	Tipos de Testes	19
3.3	Avaliando os Resultados	21
3.4	Testes de Usabilidade Remotos	22
4	Estudo de Caso	24
4.1	Introdução	24
4.2	Odontologia na UNIOESTE	24
4.3	O Sistema SISO	25
4.4	Avaliando Os Grupos de Usuários	26
4.4.1	Perfil dos Usuários do SISO	27
4.4.2	Resultados Obtidos	28
4.5	Testes de Usabilidade no SISO	29
4.5.1	O Uso da Ferramenta <i>RoboReplay</i>	29
4.5.2	Os Testes	30
4.5.3	Avaliação da Interface	30
5	Obtendo Usabilidade em Sistemas Web	32
5.1	Introdução	32
5.2	Formulários	32
5.2.1	Botões de Ação	33
5.2.2	Distribuição dos Campos	34
5.2.3	Submissão dos Dados de Formulários Através de AJAX	35
5.2.4	Tecla Enter para Submeter Formulários	37
6	Conclusão	38
A	Perfil do Usuário	40
	Referências Bibliográficas	42

Resumo

A usabilidade consiste nos mecanismos empregados em um artefato computacional ou não para que este se torne mais usável. Os sistemas Web em geral possuem uma abrangência muito maior do que os sistemas Desktop, sendo assim, as necessidades de uma diversidade maior de usuários devem ser levadas em consideração. Nesse trabalho serão apresentados os conceitos básicos sobre a usabilidade, em particular a usabilidade de sistemas Web. Esta monografia será usada como base para o desenvolvimento do SISO (Sistema Odontológico) e espera-se com isso obter um sistema que possa ser utilizado de maneira adequada por todos os membros das clínicas de odontologia da Unioeste.

Palavras-chave: Usabilidade, Sistemas Web, Sistema Odontológico, SISO.

Capítulo 1

Introdução

O uso de sistemas computacionais nas mais variadas áreas, tanto no meio científico como no comercial, é uma realidade há várias décadas. Com o barateamento e conseqüentemente a popularização dos computadores, o número de pessoas que o utilizam cresce a diariamente. Esta nova realidade insere no contexto, anteriormente apenas técnico/científico, pessoas com pouco ou nenhum conhecimento na área da informática.

Para que pessoas não ligadas a informática consigam fazer uso das ferramentas disponíveis para o auxílio em suas atividades cotidianas, é necessário um estudo aprofundado sobre como este grupo de usuários interage com a nova tecnologia.

O estudo das técnicas que visam melhorar a interação entre homens e máquinas é chamada de usabilidade. A usabilidade vem sendo empregada de muitas maneiras na sociedade muito antes da invenção dos computadores, pois ela estuda a forma como manuseamos objetos.

Segundo Nielsen [15], a usabilidade assumiu um papel muito mais importante na economia da Internet do que no passado. No desenvolvimento de produtos tradicionais o usuário apenas experimentava a usabilidade do mesmo após a sua aquisição, na Internet os sites e sistemas podem ser experimentados antes do "fechamento do negócio".

1.1 Objetivos

O objetivo geral deste trabalho é estudar os mecanismos que são utilizados durante o desenvolvimento de sistemas Web para os tornar mais usáveis.

Este projeto está inserido em um contexto maior cujo objetivo é desenvolver um sistema inovador para a automação de clínicas odontológicas de escolas de odontologia.

Como objetivos específicos tem-se:

1. Produzir uma revisão bibliográfica sobre usabilidade, em particular da usabilidade em sistemas Web;
2. Aplicar as técnicas levantadas durante a revisão bibliográfica a fim de tornar o sistema SISO usável;
3. Avaliar o grau de usabilidade do sistema através de testes com os usuários finais;
4. A partir dos conhecimentos obtidos durante os testes de usabilidade escrever um guia que possa vir a ser utilizado em outras aplicações Web.

1.2 Usabilidade

A usabilidade trata dos mecanismos empregados para tornar um artefato, computacional, ou não, adequado a seus usuários.

Não podemos atribuir apenas à interface toda a responsabilidade sobre o conhecimento que o usuário irá adquirir sobre o sistema. Conceitos muito mais intrínsecos, tais como: mapas mentais; conhecimento prévio do mini-mundo da aplicação; intelecto do usuário padrão; etc, devem ser levados em consideração durante o processo de desenvolvimento de sistemas que pretendam alcançar uma boa usabilidade.

As métricas tradicionais de avaliação de software quanto à sua usabilidade, embora consideradas eficientes, apresentam algumas lacunas quando são aplicadas em sistemas Web. Segundo Shubin [9], freqüentemente nos sistemas Web os usuários ficam confusos pois o modelo navegacional da Web é diferente do que o encontrado nas aplicações Desktop. Esta diferença é um dos elementos extras que devem ser considerados para que sistemas Web sejam usáveis.

1.3 SISO - Sistema Odontológico

A CO (Clínica Odontológica) da UNIOESTE destaca-se no contexto da área de saúde, pela sua dimensão acadêmico-social. Nas suas diversas atuações, busca estabelecer uma articulação

forte entre os três eixos de formação do acadêmico, quais sejam o ensino, a pesquisa e a extensão. De outro lado, procura contribuir para a melhoria da qualidade da saúde bucal da população da região oeste do estado do Paraná.

A CO atende acadêmicos de graduação, aperfeiçoamento e especialização. Esta possibilita aos mesmos o acesso a um conhecimento teórico acompanhado de um intenso treinamento prático, o que tem sido possível graças à integração entre alunos, professores e comunidade.

Um dos principais problemas da CO é o fato de que todos os procedimentos, tanto os de ordem administrativa (e relativos às especialidades de cada clínica que compõe a CO), quanto os de acompanhamento da realização do procedimento no paciente, são feitos manualmente. Isso faz com que haja, além de redundância de informações, a dificuldade em obtê-las. Informações básicas como definir dia e hora para atendimento, acompanhamento de listas de espera, identificação de procedimentos vinculando-os aos atendimentos financiados ou não por entidades/projetos, que hoje são dificultadas pelo controle manual, com o SISO elas passam a ser realizadas automaticamente e com maior eficiência.

1.4 Organização do Trabalho

Este trabalho está organizado da seguinte maneira: neste Capítulo 1 foi apresentada de maneira geral a contextualização e a motivação do mesmo frente as necessidades do acréscimo de usabilidade em sistemas Web.

No Capítulo 2 serão apresentados os conceitos básicos sobre sistemas Web, tecnologias utilizadas no seu desenvolvimento e o impacto que cada uma delas pode causar sobre a usabilidade.

No Capítulo 3 será apresentado um breve histórico da usabilidade em sistemas computacionais tradicionais e em seguida uma visão geral dos mecanismos empregados para aumentar a usabilidade em sistemas Web.

No Capítulo 4 será apresentado o estudo de caso, seus processos principais e metodologias utilizadas no seu desenvolvimento.

No Capítulo 5 será apresentado um guia produzido a partir da experiência adquirida durante o desenvolvimento do SISO e em seus testes de usabilidade do sistema SISO.

Na Conclusão serão apresentados os resultados obtidos tanto durante o processo de revisão bibliográfica quanto nos testes de usabilidade realizados pelos usuários finais do sistema.

As Referências Bibliográficas apresentam o material coletado durante o levantamento bibliográfico e podem vir a ser utilizadas por outros trabalhos sobre usabilidade na Web.

Capítulo 2

Sistemas Web

A Web está a cada dia mais presente nos ambientes de trabalho. A migração dos atuais sistemas administrativos do modelo Desktop para o modelo Web é um avanço natural.

Uma das maneiras de classificar sistemas computacionais é através de sua arquitetura, segundo Wixox [22], existem basicamente 3 arquiteturas:

1. Arquitetura baseada em Servidor;
2. Arquitetura baseada em Cliente;
3. Arquitetura baseada em Cliente-Servidor.

No princípio da computação os sistemas eram baseados na Arquitetura de Servidor. Nessa arquitetura todos os processos são executados em uma máquina central denominada Servidor. Aos Servidores cabem todas as tarefas, isto é, acesso a base de dados, execução da lógica de negócios e a renderização da interface. As máquinas usadas para acessar o sistema estabelecem uma conexão com o servidor e a partir desse momento elas se transformam em "terminais burros"¹

A segunda arquitetura existente foi a Arquitetura baseada em Cliente. Ela surgiu em resposta aos altos custos de servidores e *mainframes*. Nesta nova arquitetura a máquina denominada Cliente realiza todas as tarefas, processamento, lógica de negócios e em alguns casos até mesmo o armazenamento da base de dados. Esta arquitetura permite a introdução de uma máquina "pseudo-servidor" cuja tarefa é única e exclusivamente armazenar a base de dados.

¹Maquinas denominadas terminais burros não realizam nenhum processamento, elas oferecem seu hardware apenas como mecanismo de entrada e saída de dados que são enviados e recebidos do servidor.

	Servidor	Cliente
Custo Unitário	Alto	Baixo
Manutenção	Centralizada no servidor	Distribuída nos clientes
Falha no Hardware	Todos os usuários são afetados	Terminal com problema é afetado.

Tabela 2.1: Arquitetura Cliente X Arquitetura Servidor.

2.1 Arquitetura Cliente-Servidor

O modelo Cliente-Servidor é a junção da Arquitetura baseada em Cliente com a Arquitetura baseada em Servidor. Esta nova arquitetura permite que os processos genéricos a todos os clientes tais como: SGBD (Sistema Gerenciador de Banco de Dados), Servidor Web, Servidor de E-mail, etc, estejam sob responsabilidade do servidor, sendo que a apresentação da interface fica sobre competência do cliente e a lógica de negócio é balanceada entre o cliente e o servidor.

Esta nova arquitetura uniu as melhores características das suas antecessoras. Do modelo Servidor herdou a existência de uma máquina centralizadora de recursos; e do modelo Cliente a utilização dos clientes para realizar parte do processamento da lógica de negócio. Além disso, a Arquitetura Cliente-Servidor permitiu que máquinas com pouco poder computacional pudessem ser novamente utilizadas como clientes. Isso ocorreu principalmente devido à diminuição do processamento depositado sobre os clientes.

No modelo Cliente-Servidor podemos ter dois tipos de clientes.

1. Clientes Magros: são clientes que executam pouco da lógica de negócio, a maior parte do seu processamento é utilizado para a apresentação da interface do sistema;
2. Clientes Gordos ou Clientes Ricos: além da apresentação da interface do sistema, o cliente também fica responsável por uma parte da lógica de negócio.

Os Clientes Gordos geralmente são aplicativos específicos, atendem a um único sistema. Geralmente os clientes gordos são escritos em linguagens de programação como *Delphi*, *Visual Basic*, entre outras. Os Clientes Magros são freqüentemente aplicativos de propósito geral, eles podem executar diversos tipos de sistemas. Os navegadores Web são os Clientes Magros mais comuns no mercado.

2.2 Navegador Web

Os Clientes Magros usados para acessar sistemas Web são os navegadores Web. Existem no mercado inúmeros navegadores, cada um com suas características próprias. A diversidade de navegadores é um dos aspectos que deve ser considerado durante o processo de desenvolvimento da camada de apresentação de sistemas Web. A figura 2.1 apresenta estatísticas sobre o percentual de uso dos navegadores Web mais populares.

Web Browsers		
1	Internet Explorer 6.0	48.84%
2	Internet Explorer 7.0	17.64%
3	Firefox 2.0	15.45%
4	Firefox 1.5	8.29%
5	Safari 2.0	1.79%
6	Firefox 1.0	1.19%
7	Mozilla 1.8	0.68%
8	Opera 9.0	0.66%
9	AOL 6.0	0.54%
10	AOL 7.0	0.46%

Figura 2.1: Estatísticas sobre o percentual de uso dos navegadores mais populares.

A figura 2.1 foi produzida pela empresa W3counter (<http://www.w3counter.com/>) no dia 10 de julho de 2007, estes dados foram baseados nos últimos 29.866.594 visitantes únicos de 5.027 sites diferentes. Com base nesses dados pode-se com um certo grau de segurança generalizar tais estatísticas para o público geral.

Grande parte dos navegadores Web seguem os padrões definidos pelo W3C ². Os navegadores que ignoram as recomendações causam grande transtorno, tanto para os projetistas de sistemas Web, pois existe sempre a busca da portabilidade ³, quanto para os usuários finais, que podem ser impedidos de acessar algum sistema que não seja compatível com o seu navegador.

²O W3C ("World Wide Web Consortium") é um consórcio internacional formado por empresas de tecnologia que trabalham em conjunto para desenvolver os padrões que devem ser usados na Internet.

³A portabilidade de um programa de computador é a sua capacidade de ser compilado ou executado em diferentes arquiteturas, seja de hardware ou de software. O termo pode ser usado também para se referir à reescrita de um código fonte para uma outra linguagem de computador.

Quando trata-se da usabilidade em sistemas Web, é de grande importância verificar a forma como os diversos navegadores apresentam os elementos que compõem a interface do sistema. Na figura 2.2 nota-se claramente a diferença entre a apresentação do componente *select* em 3 dos navegadores mais populares (Internet Explorer, Mozilla FireFox e Opera). Esta diferença na apresentação irá diminuir a usabilidade do sistema pois no Internet Explorer grande parte da informação fica inacessível ao usuário.

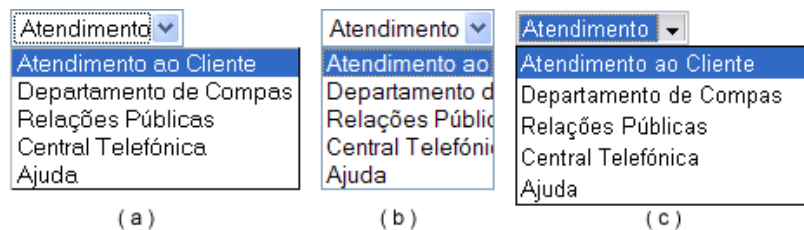


Figura 2.2: Representação de *select* nos navegadores: (a) Mozilla Firefox 2.0, (b) Internet Explorer 7.0 e (c) Opera 8.0.

Em sistemas Web onde existem frases como: "Melhor Acessado utilizando o navegador Internet Explorer 5.0 ou superior.", ou ainda, "Para o melhor uso deste sistema é necessário a instalação do *plug-in* ...", indicam falta de conformidade com o padrão estabelecido pela W3C.

"Qualquer um que inadvertidamente coloque o selo 'esta página é melhor visualizada com o navegador X' em sua pagina Web, parecem ansiar por aqueles velhos tempos, antes da Web, quando você tinha uma pequena chance de ler um documento feito em outro computador, outro processador de texto, ou outra rede." Tim Berners-Lee em Technology Review, Julho de 1996

2.3 Aplicações Ricas no Navegador Web

Com o aumento dos requisitos que os sistemas Web devem atender, ficou evidente que o navegador Web tradicional não possui mecanismos nativos adequados para a criação de sistemas computacionais sofisticados. Para contornar este problema foi necessária a criação de tecnologias que fornecessem aos Clientes Magros uma gama maior de objetos para a interação com o usuário.

Segundo Crane [4], um Cliente Rico é um programa que se comunica com um processo diferente, independente, geralmente executado em um servidor.

Os Clientes Ricos apresentam o conteúdo e interagem com o mesmo de uma forma não linear. Os sistemas Web contemporâneos criaram mecanismos para simular as funcionalidades dos Clientes Ricos em um navegador Web. Esta maneira de agir permite que os aplicativos Web desfrutem da mesma agilidade que as interfaces dos sistemas Desktop. A figura 2.3 apresenta um sistema Web cuja interface foi desenvolvido utilizando a biblioteca EXT (<http://www.extjs.com>).



Figura 2.3: Interface desenvolvida utilizando a biblioteca EXT

Existem basicamente três técnicas para adicionar aos navegadores as características de um Cliente Rico, cada uma das técnicas será apresentada a seguir.

2.3.1 *Applet Java*

Segundo Deitel [5], *applets* são programas Java embutidos em documentos HTML. Quando um navegador carrega uma página Web que contém um *applet* este é baixado para o navegador e então é executado. Os *applets* são aplicativos independentes que utilizam apenas o navegador como seu hospedeiro.

Código Fonte 2.1: 'Exemplo de um Applet Java em uma página HTML.'

```

1 <applet code='notas.class' codebase='endereço' width='10' height='10'>
2   <param name='nome_aplicacao' value='Sistema de Notas'>
3 </applet>

```

Para que um *applet* seja executado pelo navegador é necessário que um software extra esteja instalado na máquina do cliente. Assim que o *applet* é carregado pelo navegador Web a máquina virtual Java ⁴ é acionada e a partir daí todas as funções que o sistema Web utiliza são executadas por ela e não mais pelo navegador Web.

2.3.2 Flash

A primeira tentativa de transformar a Web em um ambiente 'animado' foi a criação de um formato de imagem conhecido como GIF animado. Essas imagens funcionam como mini-vídeos que repetem sucessivamente uma sequência de imagens criando a sensação de movimento.

O GIF animado não permite nenhuma interação entre a aplicação e o usuário. Explorando este nicho de mercado, a empresa Macromedia ⁵ desenvolveu uma tecnologia conhecida como Flash. O Flash é um software capaz de produzir animações que permitem interação com o usuário.

Por algum tempo a única função do Flash foi produzir animações, porém com a melhora da linguagem *Action Script*, linguagem de programação embutida no Flash, os desenvolvedores Web começaram a utilizá-lo para implementar sistemas Web.

O Flash tem como ponto forte um ambiente de desenvolvimento bastante completo que possui um número grande de componentes prontos. Outro ponto forte do Flash é a beleza das interfaces que podem ser produzidas por esta ferramenta. Como pontos negativos tem-se a necessidade da instalação do *plug-in* Flash no navegador do usuário para que este possa acessar aplicativos em Flash. Além disso, os sistemas desenvolvidos em Flash não podem ser lidos pelos navegadores em modo texto e pelos navegadores de voz utilizados pelos deficientes visuais o que os torna inacessíveis para este grupo de usuários.

Tanto o uso de *applets* Java como o uso de Flash tiram vantagem apenas de seus respectivos **plug-ins** e fazem pouco uso dos recursos nativos do navegador. A próxima técnica apresentada permite que o navegador se transforme em um Cliente Rico tirando proveito apenas dos seus

⁴Máquina virtual Java (do inglês Java Virtual Machine - JVM) é um programa que carrega e executa os aplicativos Java, convertendo os *bytecodes* em código executável de máquina. A JVM é responsável pelo gerenciamento dos aplicativos, à medida que são executados.

⁵A Macromedia é uma empresa norte americana que produz software para o desenvolvimento de aplicativos Web. Em 3 de Dezembro de 2005 foi comprada pela sua maior rival a Adobe.

próprios recursos.

2.3.3 AJAX

AJAX (*Asynchronous Javascript and XML*) é o uso padronizado de Javascript e XML (*eXtensible Markup Language*) para tornar o navegador mais interativo com o usuário, utilizando para isto as solicitações assíncronas de informações entre o navegador e o servidor.

As aplicações em AJAX, em geral, utilizam um conjunto básico de 4 tecnologias:

Javascript

É uma linguagem de programação interpretada cujos *scripts* são embutidos em arquivos HTML a fim de serem interpretados pelo navegador Web.

Segundo Sebesta [19], a interpretação permite uma fácil implementação de muitas operações de depuração de código-fonte (...) por outro lado este método de programação é de 10 a 100 vezes mais lento do que sistemas compilados.

O Javascript permite a implementação de *scripts* simples e de fácil manutenção, porém, a execução dos mesmos pelo navegador Web é bastante lenta. Sendo assim é de grande importância para a usabilidade de um sistema Web que os *scripts* em Javascript sejam bem codificados.

CSS (Cascading Style Sheet)

Segundo Hakon [11], CSS é um mecanismo simples para a adição de estilos (fonte, cor e espaçamento) a um documento Web. Já segundo Crane [4], em aplicativos Web clássicos, as folhas de estilo fornecem uma maneira útil de definir um estilo em um único lugar que pode ser reutilizado em diferentes páginas Web.

Com estas duas características pode-se notar que o CSS promove a flexibilidade e a separação dos elementos que estilizam a interface dos elementos que a compõem, isso permite entre outras coisas que a interface da aplicação seja adaptada para uma necessidade especial do usuário apenas alterando a sua folha de estilo CSS.

Código Fonte 2.2: CSS para formatar um parágrafo

```
1 <style type='text/css'>
2 .paragrafoVermelho{
3     color: #ff0000;
4 }
5 </style>
```

```

6
7 <p class='paragrafoVermelho'> Modelo utilizando CSS</p>
8
9 <font color='red'>
10     <p>Modelo Tradicional</p>
11 </font>

```

No código fonte 2.2 temos uma classe chamada "paragrafoVermelho". Supondo que esta classe esteja estilizando vários elementos HTML, para alterar o tamanho da fonte por exemplo, seria necessário alterar a definição da classe e não nos elementos que a utilizam.

Supondo que a mesma página HTML fosse acessada por um usuário que possua necessidades especiais, por exemplo, baixa definição de contraste, é possível que sejam criadas múltiplas folhas de estilo que podem ser trocadas em tempo de execução. Desta forma o próprio usuário pode escolher a folha de estilo que mais lhe agrada ou que atenda melhor suas necessidades.

O Yahoo (<http://www.yahoo.com>) foi um dos primeiros portais a empregar o CSS para a formatação do seu layout. Atualmente o Yahoo oferece vários recursos de personalização para seus visitantes, a figura 2.4 mostra o painel de configuração do Yahoo, nele é possível definir o esquema de cores e também a largura do layout.

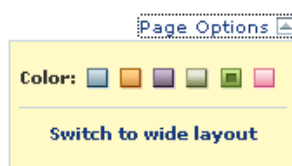


Figura 2.4: Painel de configuração de layout do Yahoo

DOM *Document Object Module*

Segundo o W3C [21], o DOM é uma plataforma neutra que permite dinamicamente que *scripts* de diversas linguagens acessem os elementos da estrutura de interface escritas em HTML.

É através do DOM que sistemas Web que utilizam AJAX conseguem atualizar porções da interface sem a necessidade de uma atualização total da página HTML. A figura 2.5 mostra o DOM de uma página HTML. A página HTML representada na figura 2.5 é composta por 3 partes, em (a) observa-se o estilo criado CSS, em (b) é mostrada a forma como um elemento

```

<html>
  <head>
    <style type="text/css">
      (a) 1
          2 .paragrafoVermelho{
          3     color: #ff0000;
          4   }
          5
    </style>
  </head>
  <body>
    (b) <p class="paragrafoVermelho"> Modelo utilizando CSS </p>
        <font color="red">
    (c) <p>Modelo Tradicional </p>
        </font>
  </body>
</html>

```

Figura 2.5: Exemplo do DOM de uma página HTML

da página informa ao navegador o estilo CSS que deve ser vinculado a ele, em (c) observa-se a forma tradicional de aplicar formatação dentro do documento HTML.

Objeto O *XMLHttpRequest*

O *XMLHttpRequest* é um objeto Javascript que torna possível a comunicação assíncrona entre o navegador Web e o servidor. A comunicação assíncrona permite que sejam trocadas mensagens entre navegador e servidor sem que haja a necessidade da recarga completa da página. O objeto *XMLHttpRequest* é hoje parte da especificação estabelecida pela W3C do DOM.

Neste Capítulo falou-se sobre sistemas Web e as tecnologias básicas para a sua construção de aplicações dinâmicas na Web. no Capítulo 3 será tratado da usabilidade em sistemas Web.

Capítulo 3

Usabilidade

Logo no surgimento dos primeiros objetos produzidos pelo homem, também surgiu a usabilidade. Na figura 3.1 se mostra que a usabilidade do objeto apresentado consistia em possuir uma extremidade arredondada proporcionando um melhor manuseio.

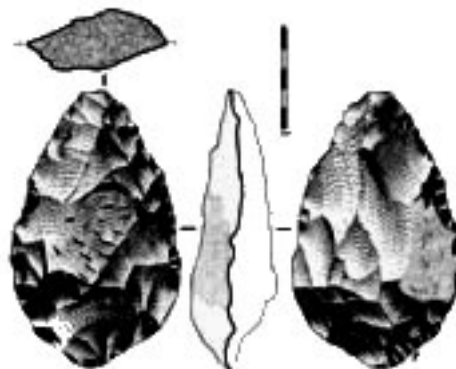


Figura 3.1: Machado manual proveniente da Espanha do período Paleolítico Inferior.

Não se deve confundir usabilidade com acessibilidade, esses dois conceitos podem ser totalmente independentes, ou seja, sistemas que tenham uma boa usabilidade podem possuir uma baixa acessibilidade e vice versa. Um exemplo deste fato são os sistemas Web desenvolvidos em Flash, neles a tecnologia proporciona recursos que podem aumentar a usabilidade, em contrapartida torna os sistemas inacessíveis aos deficientes visuais.

A usabilidade é um dos fatores que compõem a qualidade do software. Segundo a ISO [20], a qualidade do software esta fundamentada nos seguintes pontos.

- **Funcionalidade:** capacidade do software de prover funções que atendem a necessidades expressas e implícitas, quando usado nas condições especificadas;

- **Confiabilidade:** capacidade do software de manter seu nível de desempenho, quando usado nas condições especificadas;
- **Usabilidade:** capacidade do software de ser compreendido, aprendido, usado e apreciado pelo usuário, quando usado nas condições especificadas;
- **Eficiência:** capacidade do software de operar no nível de desempenho requerido, em relação à quantidade de recursos empregados, quando usado nas condições especificadas;
- **Possibilidade de manutenção :** capacidade do software de ser modificado. Modificações podem abranger correções, melhorias ou adaptações do software, mudanças de ambiente ou das especificações funcionais e de requisitos;
- **Portabilidade:** capacidade do software de ser transferido de um ambiente para outro.

3.1 O Que é Usabilidade?

A usabilidade é subjetiva pois pode variar de um indivíduo para outro. Para tentar aumentar o entendimento sobre este tema vários autores definiram a usabilidade através de abordagens diferentes. Segundo Dias [6] estas abordagens são:

- **Definição orientada ao produto:** associadas às características ergonômicas do produto;
- **Definição orientada ao usuário:** relacionadas ao esforço mental ou atitude do usuário frente ao produto;
- **Definições baseadas no desempenho do usuário:** associadas à forma de interação do usuário, com ênfase na facilidade de uso e no grau de aceitação do produto;
- **Definições orientadas ao contexto de uso :** relacionadas às tarefas específicas realizadas por usuários específicos do produto, em determinado ambiente de trabalho.

Segundo Josep [7], Usabilidade está ligada à pessoa que utiliza um produto, podendo fazê-lo de forma rápida e fácil para completar sua tarefa. Esta definição está claramente ligada ao modelo orientado ao usuário, nesse modelo, a usabilidade consiste basicamente em criar artefatos que não atrapalhem o usuário final a atingir seu objetivo.

A ISO definiu a usabilidade como sendo a capacidade de um produto ser usado por um usuário específico para atingir objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso. Esta definição mostra o aspecto subjetivo da usabilidade quando cita "...usuário específico...".

Segundo Nielsen [14], a interface dos software modernos são 48% de todo código do sistema, e consomem um terço do custo de revisão. Em contraste a engenharia da usabilidade normalmente consome apenas 6% do orçamento. Conseqüentemente melhores métodos de engenharia da usabilidade alavancarão a usabilidade do produto. Com isso observa-se que, apesar de possuir grande importância, a usabilidade ainda é pouco utilizada durante o desenvolvimento de software.

Em seu livro *Usability Engineering* [14] Nielsen descreve cinco atributos da usabilidade: facilidade de aprendizado, eficiência de uso, facilidade de memorização, baixa taxa de erros e satisfação subjetiva. Nas seções seguintes serão discutidos cada uma desses atributos.


3.1.1 Facilidade de Aprendizado

Um sistema com facilidade do aprendizado permite que usuários novatos consigam em pouco tempo, um certo grau de proficiência, para isso, o sistema deve permitir a exploração das funcionalidades do sistema e impedir, ou pelo menos minimizar, quaisquer danos acidentais causados pelo uso incorreto de alguma função do sistema. Funções de desfazer são geralmente bem vindas e muito usadas por usuários novatos, e nas ações onde não existe a opção de desfazer é necessário um aviso claro.

Uma das formas de facilitar o aprendizado é minimizando as chances de ocorrerem erros, grande parte da interação do usuário com sistemas Web se dá através de formulários. Minimizar erros de entradas de dados através da definição de máscaras de dados, como nos campos CEP e Telefone como mostra a figura 3.2 é uma maneira de aumentar a facilidade de aprendizado.

Segundo Norman [16], as propriedades físicas dos objetos limitam as operações possíveis. No campo CEP da figura 3.2 a quantidade de caracteres é limitada pela máscara do campo, facilitando assim o aprendizado.

» Endereço

* Município 

* Rua

* Número

* Bairro

CEP -

Telefone -



 Cadastrar  Cancelar

Figura 3.2: Parte do formulário de acadêmico do sistema SISO.

3.1.2 Eficiência de Uso

Uma das características da eficiência ¹ de uso pode ser descrita como a razão entre o benefício produzido com a execução da ação e o tempo gasto para aprende-la, quanto maior for esta relação mais eficiente é o sistema.

$$Eficiência = \frac{ValorProduzido}{TempoAprendizado}$$

Outra característica da eficiência do uso é o tempo de resposta, intervalo de tempo entre a solicitação do usuário e a alteração no estado do sistema.

Os sistemas Web são acessados através de redes de computadores. As redes de computadores estão sujeitas à sazonalidade de uso podendo em alguns momentos ficar congestionada. Não se pode garantir uma taxa contínua de comunicação entre o Servidor e o Cliente por isso a aplicação, deve sempre que possível, informar ao usuário a situação da sua solicitação.

3.1.3 Facilidade de Memorização

É o atributo da usabilidade que permite ao usuário que fica algum tempo sem utilizar o sistema de retomar a sua operação sem a necessidade de aprender novamente as rotinas do sistema.

Segundo Norman [16], grande parte da informação de que uma pessoa necessita para realizar uma tarefa pode residir no próprio ambiente de trabalho. O comportamento pode ser definido

¹ A eficiência refere-se à relação entre os resultados obtidos e os recursos empregados

pela combinação das informações que o usuário conhece com as informações disponíveis no ambiente.

Albert Einstein certa vez disse, não aprendo nada que não posso rapidamente consultar. Sistemas usáveis devem possuir pistas visuais para que o usuário consiga recordar como realizar uma determinada tarefa.

3.1.4 Baixa Taxa de Erros

Em sistemas com baixa taxa de erros existe um ambiente em que o usuário pode se recuperar de falhas. A figura 3.3 mostra o mecanismo utilizado pelo SISO para o tratamento de campo preenchidos de forma errônea. As falhas são tratadas antes do envio dos dados ao servidor, isso diminui o tempo de resposta e minimiza os transtornos causados pela falha.

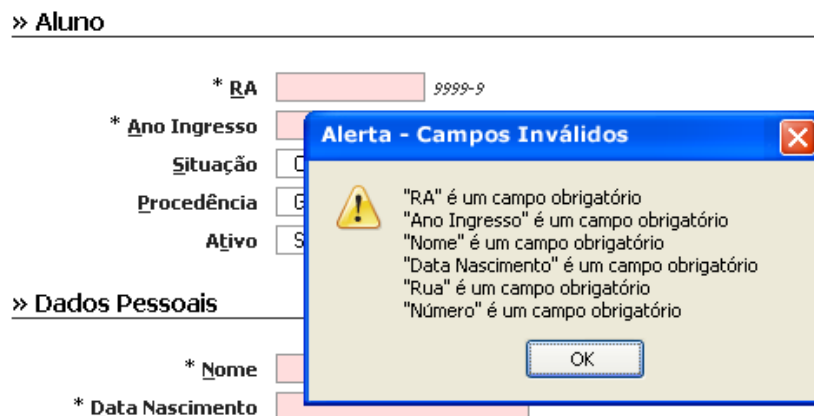


Figura 3.3: Mensagem de erro sobre falhas no preenchimento de campos do formulário.

3.1.5 Satisfação Subjetiva

A satisfação subjetiva é a propriedade da usabilidade que busca tornar o uso do sistema prazeroso. Essa característica é importante em sistemas utilizados para o lazer. Nestes casos, fatores como a eficiência do uso são relegados a segundo plano.

O Orkut (www.orkut.com) é uma rede social onde milhões de pessoas se relacionam diariamente. O Orkut não seguiu muitas das regras básicas da usabilidade. Não é eficiente pois mesmo depois de aprender a utilizá-lo os mecanismos ainda são difíceis de operar, os erros são frequentes tanto por parte do usuário como do próprio sistema pois ele foi programado usando

uma tecnologia pouco escalável ². Em oposição a tudo isto tem-se usuários com uma alta taxa de satisfação subjetiva. O sucesso do Orkut está baseado na sensação de bem-estar que seus usuários sentem em utilizá-lo.

No livro *FLOW The Psychology Of Optimal Experience*, Mihaly Csikszentmihalyi (segundo ele pronuncia-se *Chicks sent me hight*), diz que tudo gira em torno da felicidade. Qualquer objetivo na vida das pessoas seja: saúde; beleza; dinheiro ou poder, somente é válida se for trazer felicidade. Com este raciocínio ele desenvolveu a teoria do "flow", trata-se de um estado em que as pessoas estão envolvidas de tal forma com uma atividade que faz parecer que nada ao redor interessa. Neste caso, a experiência por si só é tão agradável que as pessoas realizarão a tarefa mesmo que sob um grande custo, por vontade de fazê-lo.

3.2 Testes de Usabilidade

A fim de validar a usabilidade do sistema serão realizados testes de usabilidade. Segundo Rubin [18], o teste de usabilidade é um processo no qual participantes representativos avaliam o grau que um produto se encontra em relação a critérios específicos de usabilidade. Isto significa que a seleção dos participantes deve ser criteriosa para que o perfil do usuário que realmente irá interagir com o sistema seja avaliado.

O objetivo principal do teste de usabilidade é identificar problemas no software que possam comprometer o seu uso, além disso, temos questões econômicas envolvidas. Sistemas com um alto grau de usabilidade em geral acarretam menos chamadas ao suporte técnico aumentando diretamente o lucro da empresa que o produziu.

Para cada fase do desenvolvimento do software existem tipos específicos de testes de usabilidade, alguns destes tipos serão apresentados a seguir.

3.2.1 Tipos de Testes

Existem diversas formas de avaliar a usabilidade de software, cada uma delas é adequada a uma fase do desenvolvimento do sistema. Abaixo serão apresentados os grandes grupos de testes e, em quais situações eles devem ser aplicados.

²Escalabilidade é uma característica desejável em todo o sistema, em uma rede ou em um processo, que indica sua habilidade de manipular uma porção crescente de trabalho de forma uniforme

Exploração

Este tipo de teste deve ser utilizado no início do processo de desenvolvimento. Ele busca respostas sobre quais serão as funcionalidades que o sistema deverá atender.

O Objetivo principal deste tipo de teste é explorar o modelo mental ³ do usuário a respeito da tarefa em questão. Nos testes de exploração deve haver um alto grau de interação entre o monitor e o usuário a fim de detectar pontos onde desenvolvedores e usuários divergem sobre a tarefa.

Uma abordagem que pode ser adotada para a execução deste tipo de teste é:

1. Apresentar uma tarefa do cotidiano do usuário;
2. Pedir que o usuário fale sobre a tarefa;
3. Solicitar que o usuário apresente suas expectativas sobre o que o sistema poderá fazer para facilitar a execução da tarefa;
4. Em conjunto com o usuário propor uma solução computacional.

Avaliação

Esse tipo de teste deve ser utilizado durante a fase intermediária do desenvolvimento do software. Nesses testes, o principal objetivo é identificar se os modelos mentais detectados durante os testes de exploração foram satisfatoriamente transcritos para o software.

A realização deste teste é feita com o participante executando tarefas específicas previamente estabelecidas pela equipe de testes.

Validação

Esse tipo de teste deve ser utilizado na fase final do desenvolvimento de software. Seu objetivo é certificar-se que o produto final atenda aos padrões desejados de tempo e esforço necessário para a realização de uma determinada tarefa.

³Segundo Norman [16], modelo mental é o nosso modelo conceitual particular da maneira como um objeto funciona, eventos acontecem ou pessoas se comportam, que resulta da nossa tendência de dar explicações para as coisas. Esses modelos são essenciais para nos ajudar a entender nossas experiências, prever reações de nossas ações e manipular ocorrências inesperadas. Nós baseamos nossos modelos no conhecimento que temos, real ou imaginário, ingênuo ou sofisticado.

Os analistas de usabilidade devem estimar o tempo ideal para a realização da tarefa e ao fim do teste analisar se o resultado obtido é equivalente ao que foi estimado, caso não seja deve-se revisar o tempo estabelecido ou a forma como a tarefa foi projetada pelos desenvolvedores.

Durante os testes do tipo validação deve-se ter cuidado para que não haja pressão sobre o participante pois isto pode mascarar o resultado real do teste.

Comparação

O teste de comparação pode ser aplicado em qualquer uma das fases do desenvolvimento. O objetivo dele é testar a partir de várias soluções propostas qual a mais satisfatória para a solução de um determinado problema de usabilidade.

Nesse teste o monitor deve agir de maneira semelhante à utilizada no teste de exploração, pois é de suma importância que o participante explique claramente quais foram os critérios que levaram-no à escolha de uma determinada solução.

3.3 Avaliando os Resultados

A avaliação dos resultados obtidos durante o teste de usabilidade irá nortear a equipe de desenvolvimento no processo de modificação do software. Cada problema detectado possui um grau de severidade e uma probabilidade de acontecer, as tabelas 3.1 e 3.2 abaixo foram retiradas de Ferreira [8].

Severidade	Descrição
4	Inutilizado
3	Severo
2	Moderado
1	Irritante

Tabela 3.1: Severidade do problema detectado.

Frequência	Descrição
4	+90%
3	Entre 51% e 89%
2	Entre 11% e 50%
1	Menos de 10%

Tabela 3.2: Frequência com que os problema detectado ocorre.

A tabela 3.1 representa o grau de severidade de um problema e a tabela 3.2 representa a probabilidade do mesmo acontecer. Ainda segundo Ferreira [8], pode-se ordenar os problemas através da fórmula abaixo. Os problemas que obtiverem criticidades maiores devem ser corrigidos primeiro.

$$Criticidade = Severidade + Probabilidade de Ocorrência$$

3.4 Testes de Usabilidade Remotos

Os testes de usabilidade podem ser de dois tipos, locais ou remotos. Os testes locais foram padronizados e estudados por diversos autores, geralmente são realizados com a presença de um monitor, um participante e um mediador. O ambiente é bem controlado e foi descrito pelos autores [15] e [13]. A figura 3.4, representa uma sala para testes de usabilidade.

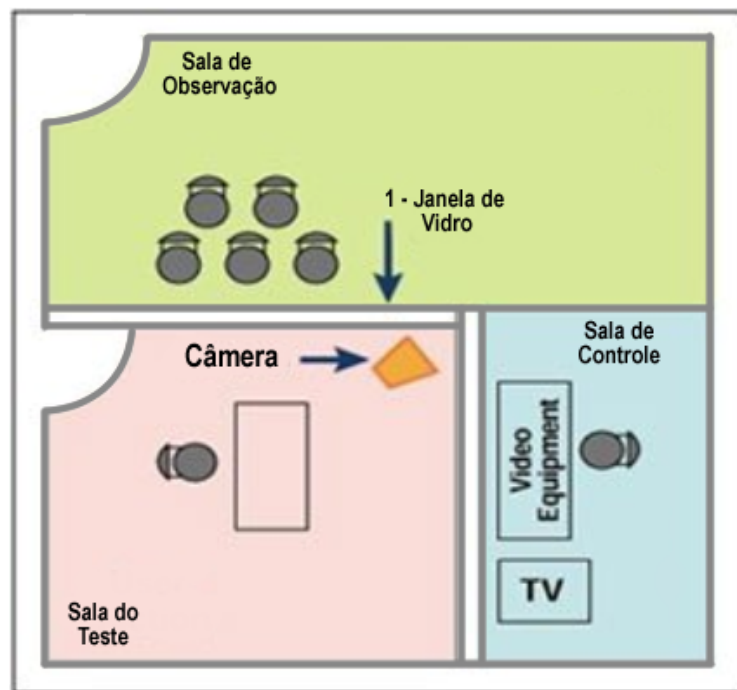


Figura 3.4: Ambiente controlado para testes de usabilidade.

A fim de baratear os testes de usabilidade foram desenvolvidos mecanismos para que os mesmos fossem realizados em ambientes distribuídos. O teste remoto vem ganhando força dentro das fábricas de software pois permite que testes de usabilidade sejam realizados com maior frequência e menor custo. Infelizmente, este tipo de teste elimina o contato direto do

monitor com o participante o que diminui a gama de percepções que podem ser avaliadas, como por exemplo expressões de alegria ou de descontentamento.

Os testes remotos podem ser aplicados tanto nos testes de avaliação, quanto nos de validação e de comparação, porém, esse tipo de teste não é aconselhado para os testes de exploração pois neste caso a figura do monitor é de suma importância para a qualidade das informações obtidas.

A realização de testes remotos requer a instalação de softwares específicos, algumas ferramentas disponíveis são WebQuilt e o RobotReplay:

- **WebQuilt:** desenvolvido por Hong [10], o WebQuilt é um sistema de *log* e visualização de atividades em sistemas Web. A finalidade deste software é ajudar equipes de desenvolvimento durante o processo de coleta de dados para avaliar a usabilidade do sistema. Para a utilização do WebQuilt é necessária a sua instalação no *proxy* que faz a ponte entre o navegador e o servidor Web. Toda interação entre eles é arquivada no *log* e posteriormente utilizada pelo WebQuilt para exibir as atividades que o participante realizou;
- **RobotReplay:** desenvolvido pela empresa nitobi, é uma ferramenta Web que pode ser encontrada no endereço (<http://www.robotreplay.com>), ela não necessita ser instalada, basta um cadastro simples no site. Essa ferramenta permite a criação de vídeos a partir da interação do visitante com o sistema Web.

O grande benefício do teste remoto é que este pode ser realizado por um número maior de usuários com custos reduzidos. Testes remotos também são bem vindos em situações onde a equipe de teste e os usuários do sistema estão geograficamente distantes.

Alguns testes de usabilidade que serão realizados no sistema SISO serão apoiados pela ferramenta RobotReplay. Esta ferramenta será utilizada durante os primeiros contatos dos usuários reais com o sistema.

Apesar de todo o esforço despendido para que as tarefas sejam usáveis, existem situações onde a falta de usabilidade é necessária. Tarefas que envolvam grandes riscos devem possuir um funcionamento diferente das demais, essa é uma maneira de diminuir a sua usabilidade.

No Capítulo 4 serão apresentados os resultados obtidos a partir dos testes de usabilidade que foram aplicados no estudo de caso deste trabalho.

Capítulo 4

Estudo de Caso

4.1 Introdução

Segundo Ceccotti [2], na sociedade atual, é um fato que a valorização da informação como ferramenta indispensável de trabalho vem obrigando, por parte dos projetistas de bancos de dados, uma mudança de comportamento bastante acentuada. Para dar conta das atuais exigências dos usuários e também das grandes empresas, os projetistas de SGBDs precisam se preocupar, cada vez mais, com o desenvolvimento de sistemas que de fato contribuam não apenas para o armazenamento dos dados nas bases e informações geradas, mas também com o próprio gerenciamento e manipulação dessas informações, conforme as demandas surgidas. É neste contexto que a Ciência da Computação pode e muito tem contribuído com soluções que gerenciem dados e informações.

Neste Capítulo será apresentado um estudo de caso. Este estudo de caso será basicamente a aplicação do conhecimento obtido durante a revisão bibliográfica no processo de desenvolvimento do sistema, e posteriormente na realização de testes de usabilidade. O SISO está sendo desenvolvido por um equipe multidisciplinar cujos integrantes são oriundos do Curso de Informática, Núcleo de Inovações Tecnológica (NIT) e do Curso de Odontologia da Unioeste.

Na primeira e segunda sessão deste Capítulo o SISO será apresentado. Na sessão 4.5.3 serão apresentados os resultados dos testes de usabilidade realizados junto aos usuários do sistema.

4.2 Odontologia na UNIOESTE

A Clínica Odontológica (CO) é referência no atendimento de todas as especialidades odontológicas, oferecendo treinamento como parte da formação em Odontologia para cerca de 40

alunos por série, bem como atende à formação continuada através de cursos de atualização, aperfeiçoamento e especialização.

A CO é composta hoje por cinco Clínicas, um Centro Cirúrgico, o Centro de Especialidades, a Clínica do Bebê e o Atendimento de Urgência, prestando um atendimento continuado, inclusive nos períodos de recesso e férias.

Neste atendimento, atualmente, estão inclusos os procedimentos de dentística básica, cirurgia básica e preventivos individuais, procedimentos de periodontia, de endodontia; de odontologia cirúrgica e traumatologia buco-maxilo-facial.

A Clínica oferece também assistência básica à saúde bucal, procedimentos que estão cobertos pelos credenciamentos junto ao SUS (Sistema Único de Saúde), na atenção básica e especializada além da participação do programa Brasil Sorridente com o credenciamento de um Centro de Especialidades Odontológicas (CEO).

Recentemente a Clínica recebeu o credenciamento do Laboratório de Prótese, significando que a produção das atividades ali desenvolvidas será financiada pelo Ministério da Saúde, através do Programa Brasil Sorridente. Portanto, considerando as especificidades do Curso de Odontologia bem como o aumento crescente das atividades desenvolvidas pela CO constatou-se a necessidade da criação e disponibilização de mecanismos para o gerenciamento das atividades ali desenvolvidas, visando superar alguns problemas que interferiam na rotina realizada.

Neste sentido surgiu a necessidade de um sistema que auxiliasse nas atividades da CO, este sistema é o SISO. O objetivo geral do SISO é contribuir para a melhoria na qualidade do atendimento administrativo oferecido pela CO aos pacientes com problemas odontológicos através da informatização das rotinas realizadas por ela.

4.3 O Sistema SISO

O SISO quando plenamente desenvolvido será um sistema Web que atenderá as necessidades de organização e sistematização das rotinas e procedimentos realizados pela CO. Para o desenvolvimento do SISO, estão sendo empregadas técnicas de processo de engenharia de software tradicionais. Portanto, os requisitos necessários ao desenvolvimento do SISO foram elicitados, analisados e atualmente estão sendo validados e em paralelo implementados.

O processo de elicitação foi feito junto aos professores, alunos de graduação e de especiali-

zação além de técnicos vinculados à CO. A análise dos requisitos foi feita através de discussões técnicas que se consolidaram na conclusão do respectivo Documento de Requisitos (DR). O DR foi aprovado por representantes da CO. A validação dos requisitos está sendo feita através dos testes e análises dos elementos e componentes do sistema com apoio da mesma equipe, ou seja, professores, alunos de graduação e de especialização assim como técnicos vinculados à CO. Os principais módulos do sistema podem ser vistos na figura 4.1

- **Paciente:** neste módulo serão manipulados os dados referentes aos pacientes;
- **Aluno:** neste módulo serão manipulados os dados referentes aos alunos;
- **SISO:** esse é o módulo de configuração do sistema, a administração da CO irá cadastrar os tipos de procedimentos que poderão ser realizados nos pacientes;
- **Plano de Tratamento:** após a chegada do paciente até a CO será feito um exame inicial e o plano de tratamento, ou seja, os procedimentos aos quais o paciente será submetido serão cadastrados no sistema.
- **Clínica:** nesse módulo serão vinculados os procedimentos, os professores e os alunos que farão parte de uma clínica;
- **Agenda e Fila:** esses módulos irão gerenciar as datas e a ordem em que os pacientes serão atendidos;
- **Dentista:** neste módulo serão manipulados os dados referentes aos professores e dentistas;

4.4 Avaliando Os Grupos de Usuários

O Sistema SISO é um sistema que atenderá uma população bem definida de usuários, sendo assim pode-se traçar com relativa facilidade o perfil dos seus usuários.

São 3 os grandes grupos de usuários que interagirão com o sistema:

- **Funcionário administrativo:** são profissionais concursados, contratados ou estagiários que trabalham no setor administrativo das clínicas. Suas funções básicas são cadastrar os pacientes e controlar a fila de espera.



Figura 4.1: Módulos principais do sistema SISO.

- **Aluno:** grande parte do volume de dados produzidos nas clínicas são provenientes dos alunos. A eles cabe a tarefa de lançar no sistema as atividades realizadas nos pacientes;
- **Professor:** representam os moderadores do sistema, são eles que validam os dados inseridos pelos alunos, além disso, são os professores que solicitam aos funcionários administrativos novos pacientes para suas clínicas.

Na próxima sessão serão apresentados os resultados do levantamento do perfil dos usuários do sistema SISO.

4.4.1 Perfil dos Usuários do SISO

No famoso livro "A Arte da Guerra" de Sun Tzu está escrito, "conheça o teu inimigo como a ti mesmo". Não se deve considerar o usuário como inimigo, porém, existe uma batalha que é tornar o sistema usável. A melhor maneira de conseguir isso é conhecendo os usuários, suas experiências anteriores, seu nível intelectual e suas expectativas sobre o sistema.

A fim de levantar o perfil dos usuários do sistema SISO, aplicou-se a uma parcela significativa da população de usuários o questionário que encontra-se no apêndice A. Ele possui questões cujo principal objetivo é responder as seguintes questões:

1. Qual é a experiência dos usuários com os computadores e a Internet?
2. Qual o navegador mais popular entre eles?

3. Os usuários já trabalharam com o sistema operacional Linux?

As duas primeiras questões são relevantes pois o SISO é um sistema Web. A última questão tem importância pois o sistema SISO será acessado através de computadores com o sistema operacional Linux.

Aproximadamente 249 usuário irão trabalhar diretamente com o SISO. A população de usuário esta distribuída da seguinte maneira: 200 alunos, 5 técnicos administrativos e 44 professores. O questionário foi aplicado 31 usuários (a aproximadamente 12% da população total). Os resultados obtidos com o questionário serão apresentado abaixo.

4.4.2 Resultados Obtidos

Com relação ao uso do computador os resultados mostram uma população altamente informatizada. Todos os usuários afirmaram conhecer e trabalhar com o computador a mais de 5 anos, além disso, 90% dos usuários possuem acesso a Internet em suas casas.

Os futuros usuários do SISO gastam em média 13,23 horas semanais no computador, sendo que em 11,77 horas a Internet pode ser acessada.

Os usuários acessam a Internet utilizando basicamente dois navegadores: Internet Explorer e Mozilla Firefox. O gráfico da figura 4.2 mostra a distribuição dos navegadores entre os usuários.

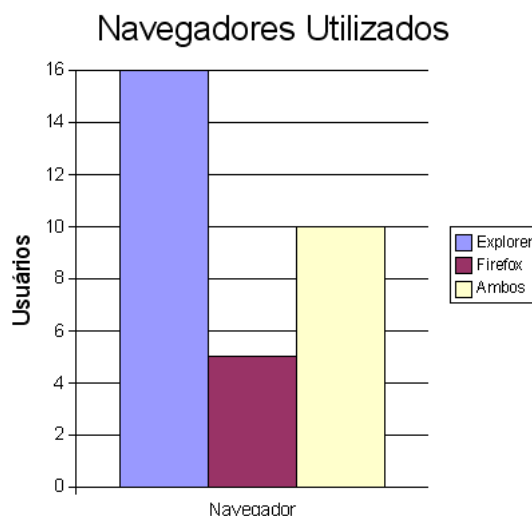


Figura 4.2: Distribuição do uso dos navegadores.

O SISO foi desenvolvido baseado em software livre, por esta razão ele foi otimizado para ser acessado através de navegadores que sigam os padrões da W3C. Dentre os navegadores que

seguem estes padrões o Firefox é o mais conhecido. Grande parte dos usuários do SISO já conhecem o Firefox o que irá facilitar no aprendizado do sistema.

A figura 4.3 traz a distribuição do conhecimento sobre Linux. Nesta figura se mostra que do total de usuários, apenas 2 já utilizaram o Linux e dos demais usuários a grande maioria o conhece porém ainda não o utilizou e, que apenas 7 usuários não conhecem o Linux.

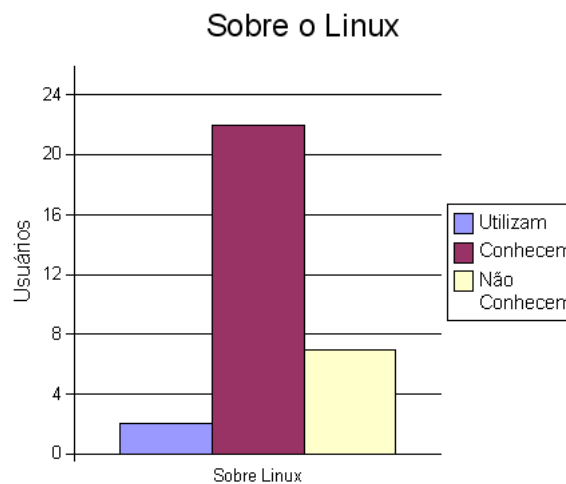


Figura 4.3: Conhecimento sobre Linux.

Levando em conta o baixo índice de usuários que já utilizaram o Linux é possível que durante os primeiros contatos com o SISO exista algum tipo de dificuldade no acesso ao sistema. Será importante um acompanhamento junto a estes usuários pois eles podem atribuir ao SISO alguma dificuldade na utilização do Linux.

4.5 Testes de Usabilidade no SISO

A fim de avaliar a usabilidade do sistema SISO alguns testes foram realizados durante o processo de desenvolvimento. A ferramenta utilizada para o acompanhamento dos testes foi o *RobotReplay*

4.5.1 O Uso da Ferramenta *RoboReplay*

Na sessão 3.4 foi apresentada a ferramenta RoboReplay. Em resumo, esta ferramenta permite que testes de usabilidade sejam realizados mesmo que avaliador e usuário estejam geograficamente distantes.

No contexto geral a ferramenta saiu-se muito bem e boa parte da interação dos usuários com o sistema puderam ser posteriormente assistidas e qualificadas pelos avaliadores.

Alguns problemas que ocorreram durante os testes devido ao uso da ferramenta foram:

1. Algumas interações mais complexas onde era utilizado AJAX não foram gravadas corretamente;
2. Existe um *delay* de dez minutos entre a interação do usuário e o vídeo estar disponível para ser avaliado;
3. Alguns vídeos não puderam ser avaliados pois os mesmos estavam tremidos.

4.5.2 Os Testes

Durante os testes de usabilidade realizados constatou-se a veracidade da premissa que norteou o desenvolvimento da interface do sistema. A premissa era de que caso a interface do sistema Web fosse semelhante às interfaces encontradas nos sistemas Desktop, isso diminuiria o impacto em utilizar um sistema Web. Esta premissa foi comprovada devido às expressões dos usuários sobre a semelhança que o sistema possuía com outras aplicações que eles já haviam utilizado.

No desenvolvimento do sistema houve a preocupação em manter o padrão da interface e das ações de cada componente. A padronização da interface e das ações aumenta a facilidade de aprendizado sendo que esta é uma das características apontadas por Nielsen [15] como pré-requisito para a obtenção da usabilidade.

4.5.3 Avaliação da Interface

O primeiro problema de usabilidade foi detectado através de uma avaliação heurística realizada pela própria equipe de desenvolvimento. O problema consistia em um texto que não era significativo quanto à sua função. Ao final do preenchimento do formulário de cadastro de acadêmico o usuário se deparava com os botões da figura 4.4 (a).

Durante a avaliação heurística um membro da equipe que não havia participado da confecção da interface clicou no botão 'Novo Acadêmico' e para sua surpresa todos os dados do acadêmico que ele estava cadastrando desapareceram. A função do botão 'Novo Acadêmico' era de limpar

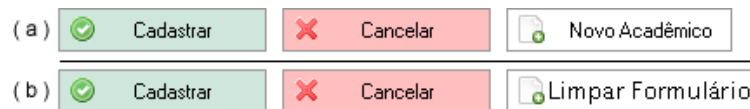


Figura 4.4: Botões de ação do formulário de cadastro de acadêmicos.

o formulário a fim de se começar um novo cadastro, ou seja, a função real era diferente da que estava no mapa mental do desenvolvedor que testava o formulário. Para solucionar este problema optou-se pelo texto 'Limpar Formulário' no lugar de 'Novo Acadêmico'. A mudança realizada na interface pode ser vista na figura 4.4 (b).

Durante os testes de usabilidade realizados junto ao usuário, foi detectada uma certa confusão por parte dos mesmos quanto às funções do 'elemento 1' e do 'elemento 2' apresentados na figura 4.5.

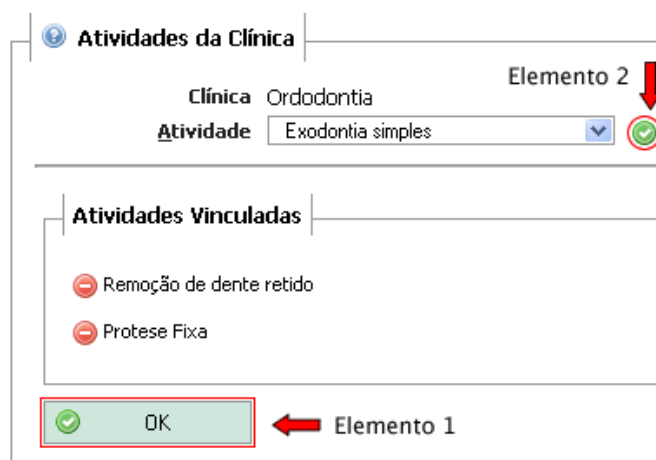


Figura 4.5: Primeiro problema de usabilidade.

Na figura 4.5 temos dois elementos com o mesmo ícone, isto fez com que os usuários não conseguissem distinguir entre eles. A solução para este problema foi a remoção do elemento 1, pois a função do mesmo era apenas sair da tela atual e ir para a tela inicial do sistema.

Após os testes de usabilidade algumas soluções adotadas no SISO foram compiladas em um guia que pode ajudar a tornar aplicações Web mais usáveis, este guia é o Capítulo 5 deste trabalho.

Capítulo 5

Obtendo Usabilidade em Sistemas Web

5.1 Introdução

Existem vários guias cujas orientações podem ser seguidas a fim de aumentar a usabilidade em Web sites, alguns guias são genéricos como por exemplo Wroblewski [23] e a referência GNOME [1], outros guias são escritos para expressar os objetivos e diretrizes de uma instituição ou organização, um exemplo deste tipo de guias é o *e-poupatempo* [17], escrito pelo governo do estado de São Paulo com o propósito de padronizar a usabilidade nos serviços Web do estado. Neste Capítulo serão feitas algumas considerações sobre como proceder para aumentar a usabilidade de sistemas Web.

5.2 Formulários

Os formulários HTML são os elementos disponíveis para que o usuário envie seus dados para a aplicação, sendo assim, é de grande importância que o uso de formulários seja intuitivo.

Em seu livro sobre HTML, Musciano [3] diz, "*Input elements should be organized in logical groups so that your brain can process the form layout in chunks of related fields*". O preenchimento de um formulário requer esforço mental e um erro no preenchimento de um campo pode ter consequências maiores do que por exemplo o clique em um link, sendo assim, faz-se necessário o estudo dos meios que diminuam a complexidade do seu uso.

Em seguida serão discutidos alguns pontos que podem ser aplicados aos formulários de sistemas Web para aumentar a sua usabilidade.

5.2.1 Botões de Ação

Em geral os formulários possuem dois botões, um primário e um secundário. O botão primário executa a ação principal do formulário, a ação que motivou o usuário a chegar até aquele ponto da aplicação, já o botão secundário pode ter duas funções, sair da tela do formulário ou a limpar os campos já preenchidos.

A distribuição e formatação dos botões representam grande parte da usabilidade do formulário pois são eles os elementos utilizados ao final do preenchimento de todos os demais campos. A escolha do botão errado por parte do usuário pode acarretar em frustração além de tempo perdido.

Na figura 5.1 são apresentadas três diferentes disposições dos botões, na abordagem (a) nota-se a semelhança entre o botão primário e o botão secundário, isso pode facilmente confundir o usuário, já nas abordagens (b) e (c) os botões além de estarem em extremos opostos do formulário, possuem formas e dizeres distintos o que previne muitos erros.

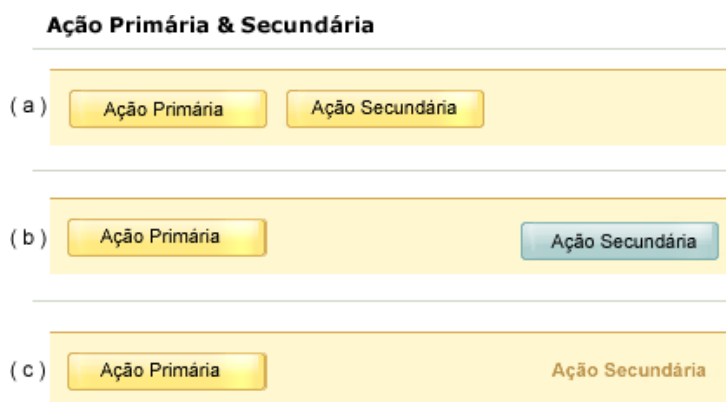


Figura 5.1: Formas de distribuir os botões em um formulário.

No sistema SISO foram adotadas as seguintes convenções para os botões dos formulários:

1. **Textos:** uso de rótulos significativos, palavras sempre padrões e consistentes, em geral as palavras adotadas foram "Salvar" e "Cancelar";
2. **Elementos visuais:** cores e ícones ajudam a identificar o botão primário e secundário;
3. **Disposição padrão:** a posição dos botões é sempre igual em todos os formulários, o botão de salvar à direita e o botão de cancelar à esquerda;

A figura 5.2 é um exemplo de formulário onde os botões são pouco usáveis, tanto por terem uma aparência semelhante quanto pela sua disposição física não ser a usual.

cancel submit

Project Name *

Project Owner

Company

Start Date

Target Finish Date

Target Budget \$

Actual Finish Date Dynamically calculated

Actual Budget \$

URL

Staging URL

cancel submit

Figura 5.2: Formulário para cadastrar projetos no DotProject.

5.2.2 Distribuição dos Campos

Levando em consideração a quantidade de campos do formulário pode-se classificá-los em 3 grupos.

- **Simples:** são formulários com poucos campos, entre 1 a 10, além disso todos os campos possuem forte relação entre si. Um exemplo deste tipo de formulário pode ser visto na figura 5.6;
- **Moderados:** são formulário que contenham entre 10 e 20 campos, além disso os campos podem ter distinção de assunto entre si, a figura 5.3 apresenta parte do formulário para cadastrar alunos, campos relacionados entre si devem ser agrupadas em conjuntos;
- **Complexo:** são formulários com mais de 20 campos tendo grande distinção de assuntos entre si, em geral este tipo de formulário é fragmentado em abas ou preenchido em etapas, estas abordagens minimizam a quantidade de campos apresentados na tela. A figura 5.4 mostra a divisão em abas do formulário de inscrição para o vestibular 2008 da Unioeste.

» **Dados Pessoais**

* Nome


* Data Nascimento

E-mail

Sexo: ☒ Masculino ☐ Feminino

CPF

» **Endereço**

* Município  *Cascavel*

* Rua

* Número

* Bairro

CEP

Telefone -

Figura 5.3: Parte do cadastro de aluno.

Uma discussão interessante sobre distribuição dos elementos de um formulário pode ser encontrada em Wroblewski [23].

5.2.3 Submissão dos Dados de Formulários Através de AJAX

Como discutido no Capítulo 2, AJAX é entre outras coisas uma maneira de navegador Web e Servidor se comunicarem de forma assíncrona. A maior parte dos dados contidos nos for-

-- 1 -- -- 2 -- -- 3 -- -- 4 -- -- 5 -- -- 6 --

*Nome:

*Sobrenome

*Sexo:

Masculino ☒

Feminino ☐

*Data Nascimento:

Figura 5.4: Ficha de inscrição para o vestibular 2008 da Unioeste.

mulários podem ser submetidos através de AJAX, com exceção dos componentes *file* ¹.

Existem dois tipos de formulários em que o uso de AJAX para a submissão dos dados é aconselhável, eles serão discutidos abaixo.

Formulário iterativos

Formulários iterativos são formulários que são preenchidos sucessivamente, e em cada novo preenchimento poucos campos sofrem alteração. Este tipo de formulário segue o diagrama de seqüência da figura 5.5.

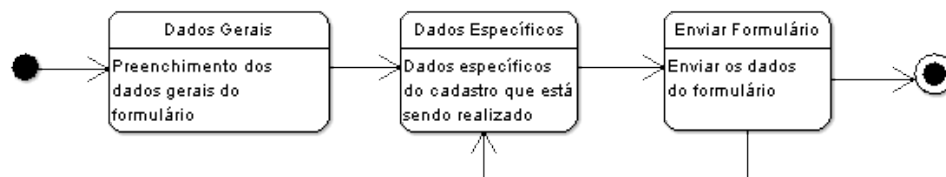


Figura 5.5: Diagramas de formulários onde o uso de AJAX é aconselhado.

O formulário da figura 5.6 pertence a este grupo. O objetivo do formulário é solicitar novos pacientes para as aulas práticas. Esse formulário deve ser preenchido para cada aluno que necessitar de novos paciente para as aulas práticas. Em cada iteração com o formulário geralmente apenas o nome do aluno é alterado, neste caso o uso de AJAX aumenta a usabilidade do formulário.

O formulário, intitulado 'Solicitar Paciente', contém três campos de seleção: 'Clínica' com a opção 'Ortodontia II' selecionada, 'Atividade' com 'Remoção de dente retido' selecionado, e 'Aluno' com 'Selecione' selecionado. Abaixo dos campos, há dois botões: 'Solicitar' (verde com ícone de checkmark) e 'Cancelar' (vermelho com ícone de X).

Figura 5.6: Formulário para solicitar paciente.

¹São campos HTML utilizados para o envio de arquivos, eles são usados por exemplo em sistema de e-mail para anexar arquivos.

Formulário longos

Algumas linguagens de programação, como PHP, possuem um tempo de sessão ². Caso o preenchimento de um formulário possa vir a levar um tempo maior do que o tempo de sessão é necessário que os dados sejam periodicamente salvos utilizando AJAX.

O uso do AJAX neste tipo de formulário é necessário pois, caso o usuário submeta os dados ao Servidor após a conexão ter sido cancelada, todos os dados preenchidos no formulário serão perdidos. Esse mecanismo foi adotado em vários projetos como por exemplo o Wordpress [http : //www.wordpress.com.br](http://www.wordpress.com.br).

5.2.4 Tecla Enter para Submeter Formulários

Em muitos formulários ao se pressionar a tecla Enter é executada a ação primária do formulário. Porém este comportamento não é padronizado.

O comportamento padrão de cada campo do formulário com relação à tecla Enter pode variar de acordo com o navegador. Abaixo será apresentado o comportamento nos três navegadores mais populares.

- **Mozilla Firefox:** a tecla Enter apenas submete os dados do formulário se o foco do cursor estiver em um elemento do tipo *input*, elementos *select* e *textarea* não submetem os dados;
- **Opera:** apenas não submete os dados quando o foco está em um componente **textarea**;
- **Internet Explorer:** possui comportamento idêntico ao do Mozilla Firefox.

Para tentar resolver este problema várias técnicas foram desenvolvidas, mas infelizmente todas elas de alguma forma acabaram inserindo novos problemas de usabilidade. A forma de minimizar este problema, porém não de eliminar, é sempre utilizar o componente *submit* como campo de envio dos dados do formulário, esta é a combinação que minimiza os problemas de usabilidade.

²Tempo de sessão é o tempo máximo que o navegador pode ficar sem se comunicar com o servidor sem que a conexão seja interrompida

Capítulo 6

Conclusão

A partir dos estudos realizados para o desenvolvimento deste trabalho, e, das observações feitas durante os testes de usabilidade do SISO, pode-se concluir que o ambiente Web é mais rico em conectividade do que o ambiente Desktop, sendo assim, é natural que a maior parte dos sistemas Desktop acabem sendo convertidos em sistemas Web.

No início do trabalho havia a idéia de que a usabilidade em sistemas Web era distinta da usabilidade em sistemas Desktop, porém, notou-se que grande parte dos conceitos já padronizados sobre a usabilidade em sistemas Desktop devem ser aplicados também aos sistemas Web.

A forma mais eficiente para mensurar a usabilidade de um produto é observando como os usuários o utilizam. Os testes de usabilidade tem este propósito e devem ser aplicados em todas as fases do desenvolvimento.

O SISO, estudo de caso utilizado no presente trabalho, trará vários benefícios aos pacientes das clínicas odontológicas da Unioeste, dentre eles podemos citar:

- Previsão de atendimento baseado nas filas de espera, figura 6.1;
- Diminui as chances do paciente ser "esquecido" na fila de espera;
- Agilidade no atendimento ao paciente pois os acadêmicos terão on-line a ficha completa sobre o paciente.

Os seguintes itens são as intenções de trabalhos futuros:

- Expandir o guia de usabilidade em sistemas Web;
- Realizar testes com outras ferramentas de testes de usabilidade remotos;

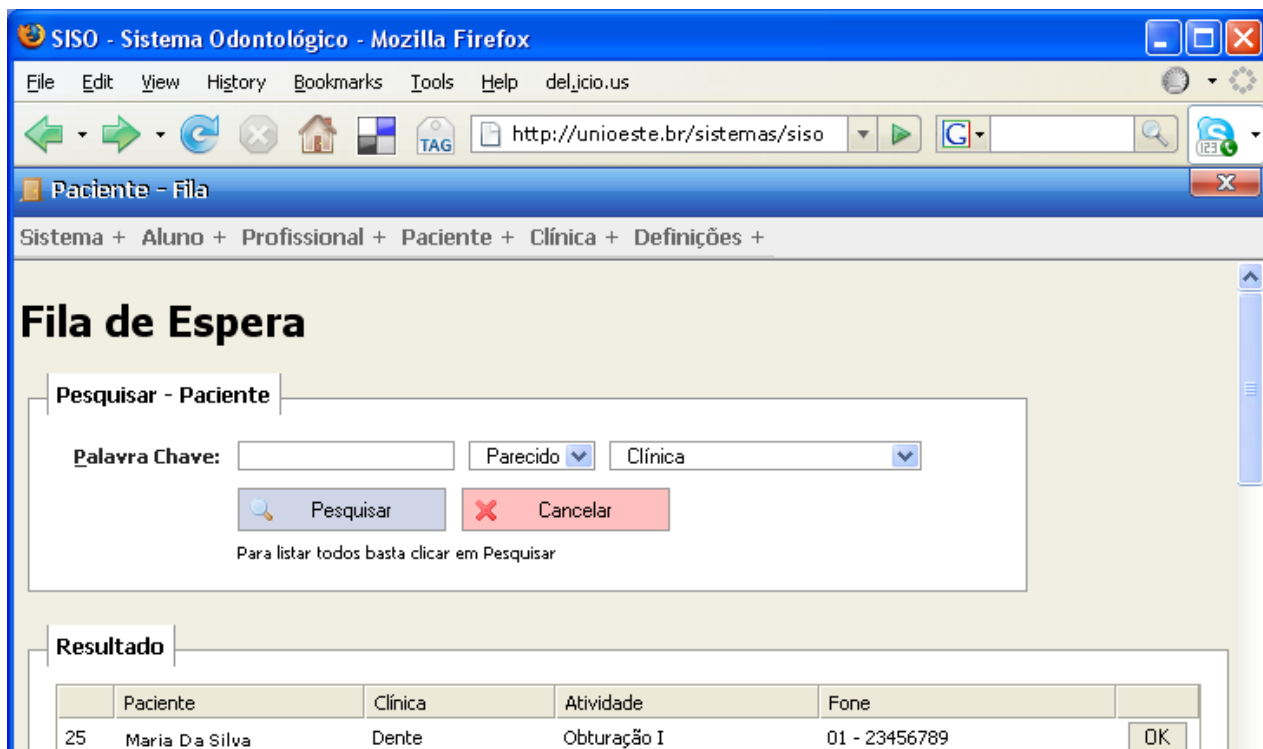


Figura 6.1: Fila de espera da clínica odontológica.

- Tornar o sistema SISO uma referência nacional em seu segmento.

Apêndice A

Perfil do Usuário

O objetivo deste formulário é obter informações sobre o perfil dos futuros usuários do sistema SISO.

I - Informações Pessoais

- 1 : Idade : [] anos
2 : Sexo : () Masculino () Feminino

II - Experiência Profissional

- 1 : Qual a sua ocupação :
(). Aluno
(). Professor
(). Funcionário Administrativo
- 2 : Há quanto tempo se encontra nesta atividade:
[] anos [] meses

III - Experiência Computacional

- 1 : Há quanto tempo você utiliza o computador:
[] anos [] meses
- 2 : Em média quantas horas por semana você utiliza o computador?
[] horas
- 3 : Há quanto tempo você utiliza a Internet?
[] anos [] meses
- 4 : Em quais lugares você utiliza a Internet?
(). Em casa
(). No trabalho
(). Na escola
(). Outros, favor especificar: _____
- 5 : Em média quantas horas por semana você utiliza a Internet :
[] horas

6 : Caso você navegue na Internet, qual navegador você utiliza, se necessário marque mais do que uma opção:

- ☐ . Internet Explore
- ☐ . Mozilla Firefox
- ☐ . Opera
- ☐ . Outro _____

7 : Já utilizou o sistema operacional Linux?

- ☐ . Sim
- ☐ . Não utilizei
- ☐ . Não conheço

Referências Bibliográficas

- [1] CALUM BENSON, E. A. **GNOME Human Interface Guidelines 2.0**. 5. ed. São Paulo - SP, 2004.
- [2] CECCOTTI, H. M. **Sistema Integrado de Informação da Área de Ortodontia da FOP/UNICAMP**. www.sibi.ufrj.br/snbu/snbu2002/oralpdf/56.a.pdf. Consultado em: 07/06/2007.
- [3] CHUCK MUSCIANO, B. K. **HTML XHTML: The Definitive Guide**. 5. ed. O'Reilly, 2002.
- [4] CRANE, D. **Ajax em Ação**. 1. ed. Rio de Janeiro - RJ: Prentice-Hall, 2007.
- [5] DEITEL, H. **Java Como Programar**. 2. ed. Rio de Janeiro - RJ: Bookman, 2002.
- [6] DIAS, C. **Usabilidade na WEB, Criando Portais mais Acessíveis**. 2. ed. Rio de Janeiro - RJ: Alta Books, 2007.
- [7] DUMAS, J.; REDISH, J. **A Practical Guide to Usability Testing**. 1. ed. Intellect Books, 1999.
- [8] FERREIRA, K. G. **Teste de Usabilidade**. 2002.
- [9] HAL, S. **Designing Web applications**. 1. ed. Washington - DC: ACM, 1997.
- [10] HONG, J. I. Webquilt: A proxy-based approach to remote web usability testing. **ACM Transactions on Information Systems**, [S.l.], v.19, n.3, p.263–285, Julho, 2001.
- [11] HAKON WIUM LIE, E. A. **Cascading Style Sheets: Designing for the Web**. 1. ed. New York - NY: Addison-Wesley, 1999.

- [12] KRUG, S. **Não Me Faça Pensar, Uma Abordagem de Bom Senso à Usabilidade na Web.** 2. ed. Rio de Janeiro - RJ: Alta Books, 2005.
- [13] MEMÓRIA, F. **Design para a Internet: Projetando a Experiência Perfeita.** 1. ed. Rio de Janeiro - RJ: Campus, 2005.
- [14] NIELSEN, J. **Usability Engineering.** 1. ed. Academic Press, 1993.
- [15] NIELSEN, J. **Projetando Websites.** 2. ed. Rio de Janeiro - RJ: Campus, 2000.
- [16] NORMAL, D. **O Design do Dia-a-Dia.** 1. ed. Rio de Janeiro - RJ: Rocco, 2006.
- [17] PAULO, G. D. S. **Manual de Usabilidade dos Serviços Públicos e-poupatempo.** 1. ed. São Paulo - SP: Imprensa Oficial, 2004.
- [18] RUBIN, J. **Handbook of Usability Testing: How to Plan, Design, and Conduct Effective Tests.** 1. ed. New York- NY: John Wiley And Son, 1994.
- [19] SEBESTA, R. W. **Conceitos de Linguagens de Programação.** 5. ed. São Paulo - SP: Bookman, 2003.
- [20] STANDARDIZATION, I. O. F. **Requisitos Ergonômicos para Trabalho de Escritórios com Computadores Parte 11:Orientações sobre Usabilidade - ISO 9241 parte 11, final draft.** 1. ed. Genebra- Suíça: ISO, 1997.
- [21] WILSON, C. **Document Object Model (DOM) Level 2 Style Specification.** Disponível em <<http://www.w3.org/TR/DOM-Level-2-Style>>.
- [22] WIXOX, D. **Análise de Projetos de Sistema.** 1. ed. Rio de Janeiro - RJ: LTC, 2005.
- [23] WROBLEWSKI, L. **Web Application Form Design.** Disponível em <<http://www.uie.com/articles/webforms/>> Consultado em: 9/10/2007.
- [24] WURMAN, R. **Ansiedade da INFORMAÇÃO 2.** 1. ed. São Paulo - SP: Cultura, 2005.