

---

# Avaliação da Usabilidade em Sistemas de Informação: o Caso do Sistema Submarino

---

Simone Bacellar Leal Ferreira  
Julio Cesar Sampaio do Prado Leite

## RESUMO

A informação de qualidade é essencial para qualquer organização e, assim, as organizações necessitam de sistemas de informação fáceis de serem usados. Essa necessidade de informação qualificada e de boa usabilidade torna a interface com o usuário parte fundamental dos sistemas de informação. Por ser a parte visível do *software*, por meio da qual os usuários se comunicam com os sistemas para executarem suas tarefas, é preciso que ela seja **amigável**, ou seja, de fácil utilização e que atenda as expectativas e necessidades de seus usuários. Para obter tais interfaces, deve ser dada atenção especial aos requisitos relacionados às entradas de dados e à exibição de informação. Apesar da obviedade das afirmações, não é freqüente na definição de sistemas de informação uma preocupação com sua usabilidade. O nosso objetivo é demonstrar por meio de uma análise de um *site* de comércio eletrônico, uma série de requisitos não funcionais que deveriam estar sempre presentes no projeto de interfaces de *software* que dão apoio a sistemas de informação.

**Palavras-chaves:** projeto de interfaces com o usuário; sistemas de informação; requisitos não funcionais; usabilidade.

## ABSTRACT

Quality information is essential organizations' success and efficacy. Because of this crucial role of information, organizations spend part of their budget with information systems. Due to the great value of information, the user interface, the way in which users interact with software to perform their tasks, becomes an important part of systems. It is necessary to design user-friendly interfaces, that is, interfaces that are easy to use. To obtain such systems' interfaces, the designers must take in consideration the non functional requirement (NFR) usability at the system definition. Failure to take NFRs in consideration has been reported in the literature. In this article we show how the lack of consideration of the NFR usability interferes on the quality of a product. We define the non-functional requirements and then analyze how some examples are used in a famous commercial Brazilian site, submarino.com. These requirements allow the designer to use facilities that improve the system's efficacy.

**Key words:** user interface design; information systems; non-functional requirements; usability.

## INTRODUÇÃO

O acesso à informação de qualidade é essencial para que uma organização obtenha a sua eficácia. Essa importância da informação é um dos fatores responsáveis pelo mercado de tecnologia da informação e pelo crescimento constante do uso da Internet. O Brasil, um dos países com as maiores taxas de crescimento de uso da Internet, possui um mercado de tecnologia da informação girando em torno de 15 bilhões de dólares por ano, sendo considerado um dos dez maiores mercados em termos globais (Ministério da Ciência e Tecnologia, 2001).

Com esse mercado crescente, as organizações começaram a desenvolver sistemas de informação baseados na Web. Esses sistemas, além de desempenharem importante papel no tratamento, processamento e distribuição da informação, estão se tornando cada dia mais importantes para as organizações, não só porque passaram a permitir uma nova modalidade de negócio, mas também por constituírem importante canal de marketing (Ferreira, 2001).

O nosso principal ponto é que a usabilidade deve ser levada em consideração em todas as interações dos sistemas que lidam com informações, principalmente para aqueles sistemas nos quais o cliente da organização interage diretamente, como no caso do comércio eletrônico (Albertin, 1999). A falta de *software* de sistemas de informação bem projetados do ponto de vista de suas interfaces com os usuários tem sido responsável por perdas de diversos tipos. Uma interface mal projetada pode causar danos graves e dificuldades na realização dos trabalhos para uma organização; mesmo assim, poucas organizações consideram esse fato quando estão planejando as suas aplicações (Laurel, 1990; Tognazzini, 1995; Shneiderman, 1997).

O nosso trabalho tem por objetivo apresentar uma taxonomia de requisitos não funcionais de usabilidade, que deve auxiliar os profissionais de sistemas de informação a identificarem e lidarem com possíveis problemas no desenvolvimento de sistemas de informação, especialmente para sistemas disponíveis na Web. Essa taxonomia foi montada com base em uma série de trabalhos já disponíveis na literatura (Foley, 1990; Laurel, 1990; Pressman, 1992; Tognazzini, 1995; Shneiderman, 1997; Nielsen, 2000) e na experiência dos autores. Exemplificamos o uso da taxonomia por meio da análise de um dos sistemas (*sites*) mais populares do Brasil, o submarino.com.br.

## REQUISITOS NÃO-FUNCIONAIS DE USABILIDADE

A usabilidade é a característica que determina se o manuseio de um produto é fácil e rapidamente aprendido, dificilmente esquecido, não provoca erros operacionais, oferece alto grau de satisfação para seus usuários e resolve eficientemente as tarefas para as quais ele foi projetado. Para garantir a usabilidade de um *site*, deve-se dar atenção aos seus requisitos não funcionais, para garantir que a informação dada ao usuário seja de qualidade (Nielsen, 2000, 2002; Pearrow, 2000; Usability Laboratory, 2001).

Um sistema orientado para a usabilidade possui uma interface que deve ser usada para se executar uma tarefa sem chamar nenhuma atenção para si, de modo a permitir que os usuários não precisem focalizar a sua energia na interface em si, mas apenas no trabalho que eles desejam executar (Norman, 1986). Isso permite que a informação flua naturalmente. Assim, os *sites* devem ser projetados visando à atender as necessidades e expectativas dos seus usuários, permitindo que eles direcionem a sua atenção para os objetos com os quais trabalham diretamente, que, por sua vez, devem refletir mais o mundo real no qual eles trabalham (Roberts, 1998).

Para que um *site* satisfaça as necessidades do usuário, o seu processo de desenvolvimento deve ser centrado no usuário, isto é, a sua interface deve ser projetada com o objetivo de satisfazer as suas necessidades e ele deve ser sempre o foco central de interesse do projetista ao longo de todo o projeto (Norman, 1986). A construção de sistemas que levam em consideração aspectos relacionados às características dos usuários e à qualidade do *software*, é um processo difícil. Várias são as razões dessa dificuldade, mas a pouca atenção dada à definição de requisitos, primeiro passo no desenvolvimento de um *software*, pode ser considerada como sendo uma das principais. Por ser a etapa inicial, ela possui forte impacto na qualidade do desenvolvimento em si e do produto final, ou na falta dela (Yeh, 1984).

Requisitos não funcionais, entre eles os de interfaces, dizem respeito à qualidade do sistema, descrevem as suas facilidades e são diretamente ligados a aspectos negligenciados da Engenharia de *Software*, que são os fatores humanos. A desconsideração desses fatores na definição de requisitos constitui uma das principais razões da insatisfação do usuário com relação a um produto (Yeh, 1984; Chung, 1995).

Os requisitos não funcionais desejáveis em uma boa interface podem ser agrupados em duas categorias: requisitos relacionados à exibição de informação

e à entrada de dados (Pressman, 1992). Abaixo detalhamos a taxonomia proposta para a análise da usabilidade de *software* para a Web. Na seção seguinte, exemplificamos o uso dessa taxonomia.

. **Requisitos Relacionados à Exibição da Informação:**

- . Consistência.
- . *Feedback*.
- . Níveis de habilidade e comportamento humano.
- . Percepção humana.
- . Metáforas.
- . Minimização de carga de memória.
- . Classificação funcional dos comandos.
- . Projeto independente da resolução do monitor.

. **Requisitos Relacionados à Entrada de Dados:**

- . Mecanismos de ajuda.
- . Prevenção de erros.
- . Tratamento de erros.

## **ANALISANDO OS RNF DE USABILIDADE DE UM SITE DE COMÉRCIO ELETRÔNICO**

### **Requisitos Relacionados à Exibição da Informação**

#### **Consistência**

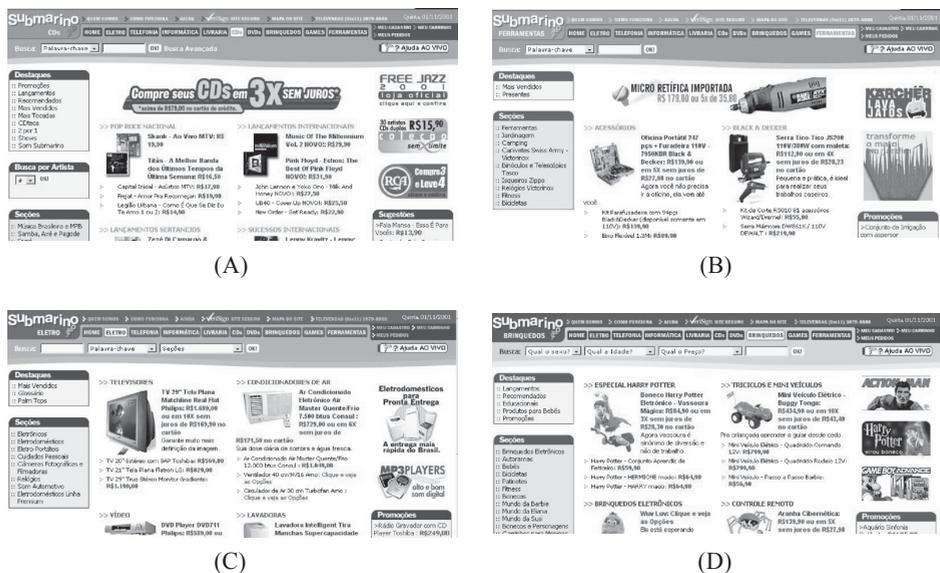
A consistência é uma das principais características para a usabilidade de uma interface (Nielsen, 2000, 2002; Pearrow, 2000). Ela reduz a frustração provocada por comportamentos inesperados e logicamente incompreensíveis do sistema. Além disso, permite que uma pessoa generalize o conhecimento de um aspecto do sistema para os outros (Foley, 1990). Para ser consistente, é necessário que

os menus, comandos de entrada, exibições de informação e todas as funções de uma interface, possuam a mesma apresentação visual e o mesmo comportamento. A Figura 1 mostra algumas telas referentes a alguns departamentos do *site* submarino.com.br. Essas interfaces apresentam inconsistências em vários de seus aspectos, merecendo destacar os seguintes:

### Apresentação Visual

Cada tela apresenta um mecanismo de busca com um formato diferente. Na Figura 1A, a busca tem início pela opção **palavra-chave**; em seguida aparece uma caixa de edição para se escolher uma busca livre e, finalmente, uma busca avançada. Já a Figura 1B não apresenta a opção de **busca avançada**. Na Figura 1C o mecanismo começa com a opção **busca livre**, depois por palavra-chave, sendo que nessa interface o usuário ainda pode pesquisar por seção. Finalmente, na Figura 1D pode-se procurar por sexo, idade, preço e busca livre. A única consistência nesses mecanismos é o botão de comando Ok, posicionado sempre no final do mecanismo de todas as telas. Outra inconsistência na Figura 1A é a troca da barra lateral direita de promoções por uma barra de sugestões.

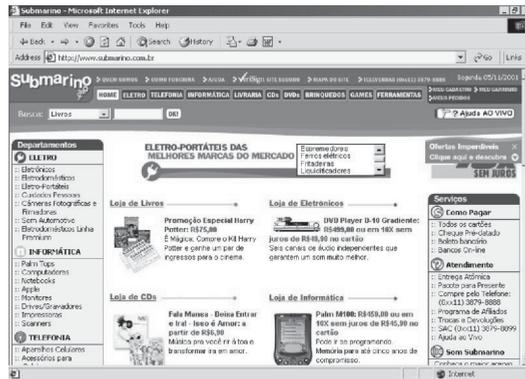
Figura 1: Interfaces do *Site* Submarino.com.br



A Figura 2 mostra a página inicial (*home page*) do *site*, cujas barras de menu horizontal apresentam suas opções (Eletro, Telefonia, Informática etc.) em ordem distinta das barras de menu verticais (Eletro, Informática, Telefonia etc.). Essa inconsistência estende-se para as imagens exibidas na tela para denotar os

departamentos, pois a ordem nas quais as imagens são mostradas difere da ordem de ambos os menus, horizontal e vertical.

Figura 2: Home Page do Site Submarino.com.br



### • Uso Adequado da Combinação de Cores

A cor, elemento fundamental em qualquer processo de comunicação, interfere nos sentidos, emoções e intelecto de uma pessoa (Marcus, 1987); ela pode ser deliberadamente usada para se atingir objetivos específicos. A combinação de cores deve ser cuidadosamente escolhida (Marcus, 1987; Jackson, 1994; Ferreira, 1999). O uso apropriado de cores pode resultar em rápida e correta assimilação da informação. O seu impacto na eficácia da interface depende da relevância do seu uso para a execução de uma tarefa e da situação e ambiente onde a tarefa ocorre (Smith, 1987). Um projetista de interface deve cuidar para não usar a cor inadequadamente, de modo a não tornar a informação dada ao usuário incompleta, ambígua ou inteligível.

O uso de cor nas interfaces do submarino.com.br é confuso e pouca ajuda o usuário. Provavelmente o projetista procurar lançar mão do uso de cores distintas para orientar o usuário na navegação pelos diversos departamentos, mas acabou exagerando, sem utilizar nenhum critério apropriado (Ferreira, 1999). A parte superior do site (vide Figura 3), com a sua marca, deveria permanecer sempre com a mesma cor; nela poderia-se colocar um *widget* do tipo fichário, com cores diferentes para cada departamento, de forma a indicar o departamento que o usuário se encontra. Em cada tela dos departamentos, essa cor da parte superior é aleatoriamente trocada, sem preocupação com relação às demais cores da tela: algumas interfaces mantêm as cores das barras laterais com a mesma cor da parte superior, outras não.

Figura 3: Parte Superior com a Marca do Site Submarino.com.br



Nota-se, porém, que em algumas partes do *site*, como no departamento de brinquedos, a cor foi adequadamente usada. Como as pessoas associam cores a diversas situações de suas vidas, pode-se fazer uso de cores para indicar diversas situações. No departamento de brinquedos foram usadas cores quentes, como vermelho e amarelo; as crianças gostam de cores vivas, pois são as principais cores de seus brinquedos (Ferreira, 1999).

### Comportamentos Inesperados

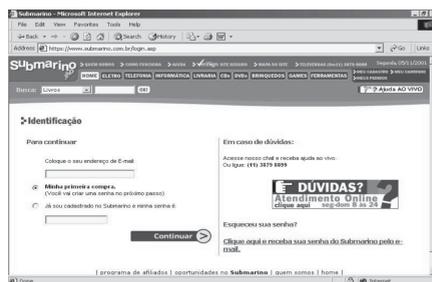
As barras laterais esquerdas de menu não possuem um padrão uniforme. Na Figura 1A aparece uma barra lateral com a opção **busca por artistas**, inexistente nas telas de outros departamentos. Naturalmente essa opção não seria pertinente a todos os departamentos; mas em alguns, como Livraria, essa opção seria adequada e, nesse caso, poderia existir uma barra lateral com um mecanismo de busca semelhante; porém no lugar de por artistas seria por autores. Isso forneceria um comportamento mais padronizado às telas de departamentos semelhantes. Para aumentar a usabilidade do *site*, ou essa barra não deveria existir, ou deveria aparecer em todas as demais nas quais esse mecanismo de busca se mostrasse pertinente.

A Figura 4 mostra outro aspecto que pode causar dúvidas no usuário. Quando a opção **Ajuda** do menu horizontal é selecionada, uma nova tela é aberta (vide Figura 4A). Certamente o usuário esperaria encontrar uma tela com o título **Ajuda**; no entanto, ele encontra uma tela com o título **Auto-Atendimento**. Da mesma forma, quando a opção **Meu Cadastro** é selecionada, é mostrada a tela intitulada **Identificação** (vide Figura 4B) no lugar de **Cadastro**. Essas pequenas sutilezas aparentemente pouco interferem na funcionalidade do *site*; mas podem ser muito significativas para a sua usabilidade.

Figura 4: Telas de Ajuda e de Cadastro



(A)



(B)

Outra falha é a opção de **Ajuda ao Vivo**, que na realidade oferece ao usuário uma tela de *chat* (vide Figura 5). Dessa forma, essa opção deveria ser denominada **Ajuda Via Chat**.

**Figura 5: Tela de Ajuda ao Vivo**



#### . **Uso de Diversos Nomes para um Mesmo Assunto**

A Figura 2 mostra que o *site* trata um único assunto por nomes distintos. Os departamentos são tratados tanto por departamentos como por lojas, como no caso de Livraria e Loja de Livros.

#### . **Uso de um Mesmo Ícone para Funções Diferentes**

Outro problema grave é o uso de um mesmo ícone para funções diferentes. A Figura 6 mostra um mesmo ícone, usado para continuar com o cadastramento de uma pessoa e para encerrar um pedido de compra. Mesmo tendo um texto com o nome da função embutido no ícone, isso pode confundir o usuário.

**Figura 6: Um Mesmo Ícone Usado para Diferentes Funções**



### ***Feedback***

Em qualquer forma de comunicação, o *feedback* é importante. Quando duas pessoas conversam, elas estão constantemente dando um *feedback* uma a outra por meio de gestos, expressões e outros. Para uma boa interação de uma pessoa

com o computador, um bom *feedback* deve ser fornecido; porém, nesse caso, ele deve ser planejado e programado (Foley, 1990).

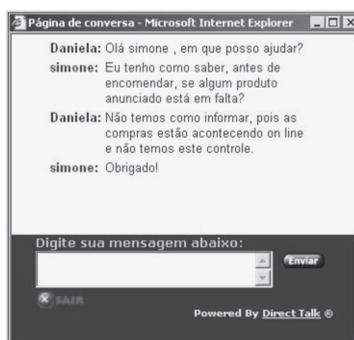
O *site* fornece alguns bons recursos de *feedbacks*, como as mensagens de erros enviadas quando algum erro é cometido. A Figura 7 mostra mensagens de erros exibidas quando o e-mail ou a senha são erroneamente escritos no momento da identificação.

**Figura 7: Mensagens de Erro**



Porém o *site* apresenta falhas quanto ao *feedback*. Se o usuário tenta comprar um produto que acabou no estoque, ele não tem como tomar conhecimento disso antes de tentar efetuar uma compra. Aparentemente esse problema teria sido resolvido por meio do recurso de ajuda (*chat*) ao vivo. Mas, ao se usar esse serviço, recebe-se a resposta de que essa informação não pode ser fornecida, uma vez que as compras acontecem *on-line* (vide Figura 8). Um melhor controle do estoque e um bom *feedback* evitariam que o usuário desperdiçasse tempo escolhendo um produto que se encontra em falta. Quando a quantidade em estoque de um produto fosse zerada, poderia ser mostrado um aviso ao lado do produto, como já é feito em outros *sites* de compras *on-line* (vide Figura 9); esta mensagem poderia até mesmo dar a opção para o usuário pedir para ser avisado por ocasião do re-estabelecimento do estoque.

**Figura 8: Feedback Fornecido pela Ajuda ao Vivo**



### Figura 9: Proposta de *Feedback* para um Produto sem Estoque



## Níveis de Habilidade e Comportamento Humano

É aconselhável que uma interface possa ser usada tanto por usuários experientes como por novatos; para isso ela deve ter algumas peculiaridades.

### . Uso de Recursos Visuais

Recursos visuais como fotografias e ícones são excelentes ferramentas para principiantes, pois os ajudam a visualizar melhor suas ações. O *site* avaliado, em sua página inicial, apresenta fotografias representativas de produtos específicos de cada loja (departamento). Isso faz com que o usuário, pelo simples olhar a tela, se dirija para a loja desejada.

Outro recurso visual utilizado é o ícone, pictograma que indica visualmente a existência de uma aplicação, ou representa uma função, um objeto, uma ação, uma propriedade ou qualquer outro conceito. Ícones bem projetados podem ser reconhecidos mais rapidamente do que palavras; se bem escolhidos, tornam-se independentes de idioma, possibilitando o uso da interface em diversos países, sem a necessidade de tradução. O *site* faz uso de ícones, porém não o suficiente; eles deixam de ser utilizados em situações nas quais poderiam ser de grande auxílio, como em algumas opções do menu horizontal (Compra Segura, Meu Carrinho, Ajuda etc.).

A Figura 10 mostra alguns ícones. Neles se percebe que não houve cuidado com a consistência no uso de letras maiúsculas e minúsculas nos textos explicativos: alguns usam só letras maiúsculas, outros misturam letras maiúsculas com minúsculas.

**Figura 10: Alguns Ícones do Site**



Um ícone bem projetado, deve ter as características abaixo.

- . Fácil Reconhecimento: rapidez com que o seu significado é captado.

**Ícones de fácil reconhecimento:** Eletro, Informática, Livraria, Brinquedos, Ferramentas, Atendimento e Ajuda ao Vivo.

**Ícones de difícil reconhecimento:** Telefonia, CDs, DVDs, *Games*, Como Pagar, Som Submarino, Galeria de Prêmios, Cultura e Diversão e Educação-on-line.

- . Fácil Recordação: rapidez com que o seu significado é lembrado, uma vez esquecido.

**Ícones de fácil recordação:** Eletro, Informática, Livraria, CDs, Ferramentas, Como Pagar, Atendimento e Ajuda ao Vivo.

**Ícones de difícil recordação:** Telefonia, DVDs, Brinquedos, *Games*, Som Submarino, Galeria de Prêmios, Cultura e Diversão e Educação-on-line.

- . Fácil Discriminação: facilidade com que um ícone pode ser distinguido de outro.

**Ícones de fácil discriminação:** Eletro, Informática, Livraria, Brinquedos, Ferramentas e Ajuda ao Vivo.

**Ícones de difícil discriminação:** Telefonia, CDs, DVDs, *Games*, Como Pagar, Atendimento, Som Submarino, Galeria de Prêmios, Cultura e Diversão e Educação-on-line.

- . **Uso de Outras Facilidades**

Facilidades como menus, formulários e *prompts* são de grande ajuda aos novatos e estão presentes ao longo de todo o *site*. Como essas facilidades muitas vezes são consideradas lentas pelos mais experientes, interfaces bem projetadas devem

permitir também o uso de aceleradores, isto é, deve permitir o uso de teclas de função e de comandos introduzidos pelo teclado, que tornam a interação mais rápida (Foley, 1990). O submarino.com.br permite que o usuário navegue por suas opções usando funções do teclado.

## Percepção Humana

A percepção de cada pessoa depende de suas habilidades para perceber e tratar as informações. Variações de habilidades físicas, comportamento e personalidade influem no êxito de um sistema. Cada usuário possui um estilo cognitivo, que determina como ele percebe a informação. Para se criar uma interface que de fato possa ser usada por diferentes pessoas, deve-se poder apresentar o seu conteúdo de diversas formas, de modo a acomodar as diferentes percepções (Pressman, 1992).

Apesar da forte tendência de se utilizar elementos gráficos no projeto de *web sites*, ainda existe muita informação na forma textual. A leitura ainda constitui atividade essencial em muitos sistemas. O tamanho do texto, o tipo de fonte, o comprimento de uma linha, as letras maiúsculas/minúsculas, a localização e cor são fatores que afetam diretamente a facilidade com a qual a informação é percebida.

O *site* submarino.com.br praticamente não pode ser personalizado, não possibilitando alterações no tamanho de suas fontes, nem via *browser*. Isso pode prejudicar a percepção de pessoas, principalmente das mais idosas, que precisam escolher fontes maiores para conseguirem visualizar o seu conteúdo.

## Metáforas

Deve-se aproveitar o conhecimento que o usuário possui a respeito do mundo que o cerca; o uso de metáforas que envolvam idéias já familiares, torna a interação mais intuitiva e menos hostil (Apple, 1992). Um dos melhores exemplos de metáforas amplamente usadas em *sites* comerciais é a do carrinho de compras. O submarino.com.br lança mão desse recurso, porém de maneira muito modesta. A Figura 11 mostra o ícone do carrinho, usado apenas no momento em que o usuário está vendo os produtos e, mesmo assim, indicando o texto **compre** e não **coloque no carrinho**. Esse ícone não é exibido na opção **meu carrinho** do menu de nenhuma tela.

**Figura 11: Uso da Metáfora Carrinho de Compras**



A Figura 12 mostra exemplos de metáforas usadas de forma inapropriada: os ícones foram usados para possibilitar ao usuário uma visualização mais rápida de algumas funções, como retirar todos os itens do carrinho e retirar um único item. Mas algumas falhas foram cometidas.

- . Não se **limpa um carrinho**, mas sim se **esvazia um carrinho**.
- . Os ícones escolhidos já são padronizados para outras funções: no caso de **Remover Item**, o ícone escolhido também é adotado nas aplicações para função de **refazer (redo)**. No caso de **Limpar Carrinho**, o ícone selecionado também é utilizado para sair de uma interface. Se o usuário tentar aplicar as heurísticas já conhecidas, ficará na dúvida sobre essas funções.

**Figura 12: Uso Inapropriado de Metáforas**



A Figura 4A mostra a tela aberta com a opção **Ajuda**. Nessa tela foi usada uma metáfora representada por uma bóia. A princípio, pode-se dizer que se trata de uma excelente metáfora; porém não se deve esquecer que a função Ajuda já é amplamente representada por um ponto de interrogação na maioria de todas as aplicações. Com isso, conclui-se que não se deve alterar o significado de itens já padronizados. Devem ser usados rótulos consistentes, abreviações padronizadas e cores previsíveis. Representações novas devem ser criadas, desde que ainda não tenham sido padronizadas; neste caso, elas não devem ser representadas por signos arbitrários; pelo contrário, devem ser cuidadosamente escolhidas, de modo que seus significados sejam claros.

## **Minimização de Carga de Memória**

Os usuários do submarino.com.br precisam de um esforço relativamente grande

para memorizar aspectos do *site*; poucos comandos mnemônicos são usados e os ícones não são tão bem projetados, o que torna a memorização não trivial (Foley, 1990). Sendo os signos (ícones, comandos textuais etc.) os elementos essenciais de uma tela, eles devem ser bem produzidos. Durante todo processo de desenvolvimento, deve ser mantida uma preocupação com a escolha e *design* dos signos, de modo que eles não induzam dúvidas e tornem a interface a mais poderosa possível (Pressman, 1992). Foram vistos diversos exemplos de ícones omitidos, mas que poderiam ter sido usados, ou de ícones que não expressam bem seus objetivos. Da mesma forma, alguns nomes de comandos não foram apropriadamente escolhidos (como limpar carrinho), o que gera dificuldade para serem memorizados.

## Classificação Funcional dos Comandos

As barras de menu fornecem uma lista de opções para o usuário; consistem de uma boa maneira para uma pessoa acessar funções não constantemente solicitadas. Seu conteúdo depende do *site*, mas, geralmente, as diversas páginas de um *site* possuem barras semelhantes, com seus itens dispostos um ao lado do outro, no caso de menu horizontal, ou um abaixo do outro, no caso de menu vertical. Um item de um menu horizontal, ao ser clicado, pode mostrar um submenu abaixo dele (menu *pull-down* ou menus hierárquicos); nesse caso, os nomes dos itens são posicionados um abaixo do outro. Uma das vantagens de um menu *pull-down* é que ele só é chamado quando necessário, economizando assim espaço de tela, sem a poluir com uma série de opções, que podem confundir o usuário.

O termo **menu** é uma metáfora com os cardápios de restaurantes e segue a sua idéia. Normalmente, em um cardápio, as opções são agrupadas abaixo de um título (aves, frutos do mar, massas etc.). Da mesma forma, em um menu de uma interface, as opções são agrupadas seguindo algum critério estabelecido pelo projetista, geralmente aquele de agrupar conforme a funcionalidade. Quando as pessoas precisam lidar com quantidade de itens, sabe-se que elas se sentem mais confortáveis se o número de itens envolvidos for, no máximo, com um valor sete, mais ou menos dois. Assim, um menu não deve ter mais do que sete, mais ou menos dois itens.

O submarino.com.br não fez uso de menus *pull-down* abaixo do seu menu principal horizontal e, com isso, acaba usando mais do que sete, mais ou menos dois itens, o que, de certa forma, polui visualmente a tela. Na realidade, quando o usuário seleciona um item do menu principal, como, por exemplo, Ferramentas, é aberta uma nova janela, que exhibe os itens dos submenus (que na tela correspondem

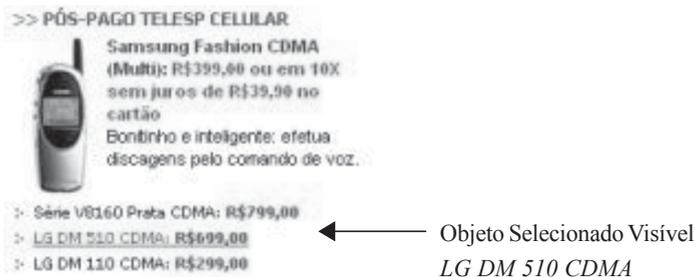
às seções jardinagem, *camping* etc.) como um menu vertical. Seria interessante se os itens do novo menu vertical fossem exibidos como menus *pull-down* e, quando o mouse passasse por cima de cada item do menu principal, eles fossem então mostrados, sem ser necessário abrir uma nova janela. Dessa forma, o usuário teria uma visão mais rápida das diversas seções, sem ter que passar por uma página intermediária desnecessariamente. Além disso, alguns itens que aparecem no menu poderiam ser agrupados com subitens do menu principal, de modo a reduzir o número de itens mostrados simultaneamente.

### . Manipulação Direta

A manipulação direta dá às pessoas a sensação de que estão controlando os objetos representados no computador; o objeto manipulado deve permanecer visível, enquanto sobre ele estiver sendo realizada uma operação e o efeito dessa operação deve ser imediatamente notado (Apple, 1992). O *site* estudado destaca bem os objetos selecionados. Quando o *mouse* passa por cima de algum item de menu, ele é destacado (vide Figura 13).

A Figura 13 mostra o item LG DM 510 CDMA selecionado e destacado.

**Figura 13: A Visibilidade dos Objetos Manipulados Deve Ser Constante**



### . Exibição apenas da Informação Relevante ao Contexto

Para ser assimilada com mais facilidade, apenas a informação relevante ao contexto corrente deve ser mostrada; o usuário não deve ter que ficar procurando, no meio de muitos dados, o que precisa para executar a sua tarefa (Pressman, 1992). Como já foi visto na Figura 1, o *site* apresenta em cada tela apenas as informações relevantes ao contexto de cada interface.

Para facilitar a assimilação, devem ser usadas, sempre que possível, janelas

distintas para mostrar informações de diferentes tipos e pelo menos uma parte de cada janela deve poder ser vista. O *site* analisado atende bem a esse requisito. Quando o usuário acessa a revista submarino, que na realidade é outro tipo de informação, ela é aberta em uma nova janela, conforme mostrado na Figura 14.

**Figura 14: Uso de Janelas Distintas para Exibição de Diferentes Informações**



## Projeto Independente da Resolução do Monitor

Outro problema que interfere na usabilidade dos *sites*, é a questão da resolução. Em interfaces tradicionais, o projetista sabe para qual ambiente ele está projetando; ele tem total controle sobre cada *pixel* da tela que aparece para o usuário, e pode ter certeza que cada elemento projetado será visualizado sempre da mesma forma nas telas, independentemente da resolução do seu monitor. O *designer* pode escolher adequadamente os aspectos visuais da interface, como tipo de fonte, tamanho, entre outros.

Como na *web*, o projetista não controla como a aparência da interface será exibida; uma vez que o usuário pode acessar a Internet de diversas maneiras, o *design* para *web* deve ser adequadamente planejado. Uma dos princípios básicos de se construir *sites* independentes da resolução, é não projetar os elementos da interface com uma largura de tamanho fixo, determinada por certo número de *pixels*; deve-se definir a aparência dos componentes por meio de porcentagens do espaço disponível (Nielsen, 2000).

Como muitas pessoas e organizações, especialmente em países não desenvolvidos, ainda possuem monitores de baixa resolução, esse aspecto deve ser levado em consideração. A Figura 15 mostra o *site* submarino.com.br exibido em um monitor de baixa resolução. Nota-se que o seu projeto não foi desenvolvido para ser independente da resolução; nessa figura, nem todos os elementos aparecem na interface, que, por sua vez, têm a sua aparência comprometida.

Figura 15: Interface do Site Exibida em um Vídeo de Baixa Resolução



## Requisitos Relacionados à Entrada de Dados

Muito tempo de trabalho do usuário é gasto com a escolha de comandos, digitação de dados e outros *inputs*. Uma boa interface deve otimizar ao máximo o tempo que o usuário gasta com essas tarefas. As diretrizes a seguir tornam a interface mais poderosa no que diz respeito à entrada de dados (Pressman, 1992).

## Mecanismos de Ajuda

Deve ser fornecida informação de ajuda para toda ação de entrada (Foley, 1990). Apesar da opção **Termos e Símbolos** do menu lateral da tela de **Ajuda** (vide Figura 4A) abrir outra janela com uma explicação dos símbolos e termos usados, o *site* não oferece dicas quando um usuário passa com o *mouse* sobre algum item. Essa facilidade deveria ter sido implementada, pois possibilita que o usuário descubra para que serve determinado item, sem ter que ir ao sistema de ajuda textual. O *site* apresenta outras facilidades de ajuda. Um exemplo desse recurso é mostrado na Figura 16, na qual, caso o usuário deseje, na hora de se cadastrar, ele pode encontrar o seu CEP selecionando uma opção.

## Prevenção de Erros

Um dos objetivos de uma boa interface é evitar que os seus usuários cometam erros. Uma interface bem projetada deve prover mecanismos de prevenção de erro, de modo que o usuário não escolha uma opção inválida para só receber uma mensagem de erro (Foley, 1990). O *site* apresenta alguns desses mecanismos.

## . Inibição de Itens Não Válidos

Deve-se inibir itens do menu que não podem ser selecionados no estado corrente do sistema. O submarino.com implementou esse mecanismo: quando um item é retirado do carrinho, o ícone de Retirar Item (vide Figura 11) é imediatamente inibido. Da mesma forma, quando o carrinho é todo esvaziado, a tela de **carrinho** é substituída por outra com a mensagem Nenhum Produto Foi Selecionado.

## . Orientação para Entrada Correta de Dados

Quando o número ou tipo de caracteres a ser entrada for limitado, deve-se indicar ao usuário antes dele digitar o texto. A Figura 16 mostra uma série de informações úteis que ajudam o usuário a não cometer erros:

- . indica a obrigatoriedade de CPF para entregas no Brasil;
- . informa que CPF só pode ser formado por números;
- . informa o número mínimo de caracteres para a senha.

**Figura 16: Indicações Úteis para o Usuário Entrar Corretamente com os Dados**

A imagem mostra um formulário de cadastro de usuário com os seguintes campos e dicas:

- Nome:** Campo de texto.
- Sobrenome:** Campo de texto.
- CPF/CNPJ:** Campo de texto com a dica "CPF obrigatório para endereços de entrega no Brasil. [Clique aqui e veja o post Q&A.](#) (somente números)".
- Endereço:** Campo de texto com a dica "nº".
- Complemento (op. 1234):** Campo de texto.
- Bairro:** Campo de texto.
- Cidade:** Campo de texto.
- Estado:** Menu suspenso com a dica "(Pedidos Internacionais)".
- Estado / Província:** Campo de texto.
- CEP:** Campo de texto com a dica "Gostaria de consultar o CEP desse endereço?".
- País:** Menu suspenso com o valor "Brasil".
- Telefone:** Campo de texto.
- Data de Nascimento (dd/mm/aaaa):** Campos de texto para dia, mês e ano.
- Sexo:** Menu suspenso.
- E-mail:** Campo de texto com o valor "lealferreira@bmcjr.br".
- Senha (mínimo de 6 caracteres):** Campo de texto com a dica "6".
- Repita a Senha:** Campo de texto.

Na base do formulário, há uma caixa de seleção marcada com o texto: "Desejo receber por e-mail informações sobre promoções e produtos do Submarino".

## . Minimizar a Quantidade de Dado a Ser Entrada

Uma boa interface minimiza o número de ações necessárias para a entrada de dados, reduzindo a tarefa do usuário de digitar. A Figura 16 mostra algumas facilidades do *site*:

- . o endereço eletrônico preenchido na tela anterior é automaticamente exibido, sem ser preciso que o usuário preencha de novo essa informação;

- . o sexo e o Estado são escolhidos a partir de uma **lista de opções** (*listboxes*), onde o usuário simplesmente seleciona a(s) letra(s) correspondente(s) a sua escolha. Além de diminuir o número de ações de entrada, esse recurso reduz a incidência de erros, pois, ao selecionar de uma lista, o usuário não tem como se distrair e escrever uma opção não válida.
- . como o *site* submarino.com.br foi projetado para residentes no Brasil, o *default* do país é definido como sendo **Brasil**, uma vez que a grande maioria das pessoas que o acessam moram no país.

### . **Interação Flexível**

Uma interface bem projetada deve permitir que o usuário controle o fluxo interativo; ele deve ser capaz de pular ações consideradas desnecessárias, alterar a ordem das ações e tratar os erros sem precisar sair do programa. O *site* analisado permite essa flexibilidade: o usuário pode colocar os produtos no carrinho, continuar comprando, retirar produtos, guardar suas compras para mais tarde e assim por diante, independentemente da ordem. Mas a interação, tipicamente com o estilo icônico ou de apontar-escolher, é pouco flexível e não pode ser voltada ao modo preferencial de entrada de dados do usuário. O usuário não tem liberdade para escolher um outro estilo.

### . **Customização**

Uma boa interface deve permitir que o usuário **customize** seus comandos e mensagens. Esse recurso não é fornecido pelo submarino.com.br. Conforme anteriormente, o usuário pouco pode customizar. Até a escolha de fontes com tamanhos distintos não é permitida.

## **Tratamento de Erros**

Todos cometem erros. Uma boa interface tem que ser capaz de permitir a sua correção o mais rápido possível; isso torna as pessoas mais produtivas e elas se sentem encorajadas a explorar o sistema, o que é uma maneira bem eficiente de se aprender as características do programa (Foley, 1990). Basicamente existem dois tipos de erros: o funcional e o sintático.

- . O erro sintático se dá quando uma seqüência de comandos (ou um único comando) é fornecida com parâmetros ou nomes errados; nesse caso, deve existir uma mensagem informativa e clara. A Figura 7 ilustra exemplo de mensagens enviadas em caso de erros sintáticos.

- . O erro funcional é o mais grave; ele ocorre quando um comando é acionado por engano, gerando resultados antecipados ou inesperados. O *site* apresenta algumas maneiras de se corrigir tais erros, como, por exemplo, retirar itens postos por engano no carrinho de compra, limpando todo o carrinho.

## CONCLUSÃO

Como é por meio da interface com o usuário que a comunicação entre o homem e um sistema (*site*) é estabelecida, o seu projeto deve fazer com que tal interação seja transparente; isto é, ao ser usada para se executar uma tarefa, a interface deve garantir que o usuário só precise focalizar a sua atenção no trabalho que ele quer executar. Uma boa interface deve satisfazer as necessidades do usuário; para isso, o seu processo de desenvolvimento deve ser centrado no usuário, ou seja, a sua interface deve ser projetada com o objetivo de satisfazer as necessidades do usuário.

Esse trabalho, a partir da apresentação de uma taxonomia de requisitos não funcionais de usabilidade e sua aplicação a um caso real, procurou mostrar que a atenção dada à definição de requisitos não funcionais pode ter forte impacto na qualidade do sistema de informação, principalmente naqueles que apresentam *software* de iteração direta com clientes da organização. A não observação da taxonomia apresentada certamente onerará a empresa em custos que poderiam ser evitados. A nossa contribuição, portanto, é a de explicitar um conjunto de orientações já disponíveis na literatura e exemplificar o seu impacto por meio da análise de um caso real. Esperamos que o nosso artigo sirva de alerta aos acadêmicos e profissionais da área de sistemas de informação para os aspectos de qualidade em relação à usabilidade, principalmente quando o usuário é externo a organização.

No entanto, vale observar que para ser de fato bem aceito, um sistema deve ser capaz de permitir que os usuários usem todos ou parte de seus recursos, de diferentes formas; isto é, o usuário deve poder escolher o que deseja usar. Aqui tratamos desse aspecto, customização, de modo superficial. Portanto justifica-se que trabalhos futuros procurem otimizar o uso de *software* de construção de *sites* que já dispõem de operadores que facilitam a customização. Verifica-se que, na realidade, as características de usuários são bastante diversificadas, sendo necessário que a análise de usuário apresente diversos modelos e enfatize a necessidade de interfaces com aspectos distintos para cada um deles, e que a interface a ser projetada possa ser usada de forma distinta pelos diversos grupos de pessoas.

## REFERÊNCIAS BIBLIOGRÁFICAS

ALBERTIN, A. L.

**Comércio eletrônico:** modelos, aspectos e contribuições de sua aplicação. 3. ed. São Paulo: Atlas, 1999.

APPLE COMPUTER.

**Macintosh human interface guidelines.** New York: Addison-Wesley, 1992.

CHUNG, L.;

NIXON, B. A.;

YU, E.

Using non-functional requirements to systematically support change. In: IEEE INTERNATIONAL SIMPOSIUM ON REQUIREMENTS ENGINEERING, 2., 1995, [S.1.]. **Proceedings...** [S.1.]: IEEE Computer Society Press, 1995. p. 132-139.

FERREIRA, S. B. L. et al.

Requisitos não funcionais para interfaces com o usuário: o uso de cores. In: WORKSHOP IBEROAMERICANO DE INGENIERÍA DE REQUISITOS Y AMBIENTES SOFTWARE, 2., 1999, [S.1.]. **Proceedings...** [S.1.: s.n.], 1999. p. 279-291.

FERREIRA, S. B. L.;

LEITE, J. C. S. P.

Usability oriented information

systems. In: BUSINESS ASSOCIATION OF LATIN AMERICAN STUDIES, 2002, [S.1.]. **Proceedings...** [S.1.]: BALAS, 2002.

FOLEY, J. D. et al.

**Computer graphics, principles and practice.** New York: Addison-Wesley, 1990.

JACKSON, R.;

MACDONALD, L.;

FREEMAN, K.

**Computer generated color:** a practical guide to presentation and display. New York: John Wiley and Sons, 1994.

LAUREL, B.

**Art of human-computer interface design.** New York: Addison-Wesley, 1990.

MARCUS, A.

Color: a tool for computer graphics communication. In: **Color in computer graphics.** California, SIGGRAPH, 1987. Tutorial notes, n. 24, p. 3-9.

MINISTÉRIO DA CIÊNCIA E TECNOLOGIA.

Disponível em: <<http://www.mct.gov.br/index.asp>>

Acesso em: 05 nov. 2001.

NIELSEN, J.

**Designing web usability.**  
Indianapolis: News Riders  
Publishing, 2000.

NIELSEN, J.;

TAHIR, M.

**Homepage:** usabilidade - 50  
websites desconstruídos. Rio de  
Janeiro: Campus, 2002.

NORMAN, D. A.

**User centered systems design.**  
New York: Lawrence Earlbaum  
Associates, 1986.

PEARROW, M.

**Web site usability handbook.**  
Massachusetts: Charles River  
Media, 2000.

PRESSMAN, R. S.

**Software engineering: a  
practioner's approach.** 3. ed. New  
York: McGraw-Hill, 1992.

ROBERTS, D. et al.

**Designing for the user with  
OVID:** bridging user interface  
design and software engineering.  
New York: MacMillan, 1998.

SHNEIDERMAN, B.

**Designing the user interface:**  
strategies for effective human-  
computer interaction. New York:  
Addison-Wesley, 1997.

SMITH, W.

Computer color: psychophysics,  
task application and aesthetics. In:  
**Color in Computer Graphics.**  
California, SIGGRAPH, 1987.  
Tutorial notes, n. 24, p. 25-27.

TOGNAZZINI, B.

**Tog on software design.** New  
York: Addison-Wesley, 1995.

USABILITY LABORATORY.

Disponível em: <[http://  
www.oslo.sintef.no/avd/32/3270/  
brosjyrer/engelsk/index.html](http://www.oslo.sintef.no/avd/32/3270/brosjyrer/engelsk/index.html)>  
Acesso em: 10 jun. 2001.

YEH, R. T. et al.

Software requirements: new  
directions and perspectives. In:  
VICK; RAMAMOORTHY.  
**Handbook of software  
engineering.** [S.l.]: Van Nostrand  
Reinhold, 1984.