



# A melhor interação entre o homem e a máquina

**ANDRÉ LUÍS BELINI DE OLIVEIRA**

Gerente de Tecnologia de Informação da Alleato Assessoria e Consultoria em TI



**O** contato com aparelhos e ferramentas interativas ocorrem o tempo todo com praticamente todas as pessoas, embora, em alguns casos, esse contato acabe por passar sem ser notado. Vão desde equipamentos de fácil manuseio até alguns mais complexos, que requerem alguns minutos para que suas reais funcionalidades sejam entendidas. Esses aparelhos estão cada vez mais presentes no dia a dia de todas as pessoas, na forma de controles remotos, aparelhos de DVD, aparelhos de som, carros com computador de bordo, sistemas de identificação automática e registro de presença, como por exemplo, no trabalho, na faculdade, no consultório médico etc.

Entretanto, nem sempre a utilização desses equipamentos ocorre de forma simples. Pode até o ser para pessoas que já possuam um prévio conhecimento ou habilidade, porém, para algumas isso se torna um objeto inibidor e em muitos casos, desestimulador, fazendo com que tarefas rotineiras, que poderiam ser mais simples, desde que o aparelho ou software em questão também fosse de fácil manuseio, continuem a ser feitas

de formas demoradas e burocráticas, criando um contrasenso entre a evolução tecnológica e sua aplicação na sociedade.

Nesse trabalho foram abordados técnicas e métodos que possam auxiliar no processo de desenvolvimento de uma interface. Foram analisados os princípios de *design* de interação e usabilidade e as consequências de sua utilização no resultado final de um projeto. Ao final desse trabalho, foi demonstrado um estudo de caso que ilustra e confirma o objeto de estudo desse artigo.

Que fatores podem ser considerados ao avaliar a funcionalidade e usabilidade de um determinado produto? Como qualificá-lo como sendo bom ou não? Esse tema vem se tornando objeto de estudo de vários profissionais de várias áreas, como desenvolvedores, projetistas, analistas de sistemas e de design, além de profissionais de IHC (Interação Humana Computador). Os profissionais da área da tecnologia da informação e comunicação começam a perceber que um dos principais fatores para o sucesso ou não dos produtos não está mais somente relacionado a um bom conheci-

mento técnico e ao uso de determinadas ferramentas de desenvolvimento.

Hoje é necessário muito mais que isso, é necessário ter clientes satisfeitos e, por outro lado, como conseguir essa satisfação? Como saber se o produto está agradando ao mercado consumidor? A interação com o sistema é agradável e fácil? As mensagens são esclarecedoras e o fluxo dentro do sistema acontece de forma intuitiva? A disposição dos objetos na tela ocorre de forma lógica e coerente ao que realmente o usuário precisa? Se as respostas forem satisfatórias, conclui-se que o produto é de fácil utilização, porém, caso isso não seja verdade, algumas alterações certamente serão necessárias.

*Design* de interação pode ser definido como um conjunto de soluções que tem como objetivo facilitar a forma como as pessoas trabalham, se comunicam e interagem. A aplicação de princípios de *design* em um *software* visa tornar o ambiente de trabalho mais funcional, objetivo e com isso, mais simples.

Uma comparação que vale a pena ser citada é a de Terry Winograd, narrada no livro *Design de Interação* –

*Além da Interação homem-computador*, de Jennifer Preece, Yvone Rogers e Helen Sharp (2005, p. 28), pois facilita o entendimento desses princípios. Diz Terry na entrevista concedida aos autores do citado livro que, para entender a diferença entre o *design* e a engenharia de *software*, bastaria que se comparasse esta com a arquitetura e a engenharia. O arquiteto busca adequar os espaços, viabilizando meios mais cômodos e agradáveis de se conviver dentro do ambiente, distribuindo os cômodos de uma casa de forma que os ambientes fiquem em harmonia e com isso torne a permanência dentro da casa agradável. Já o engenheiro cuida dos detalhes técnicos, da parte estrutural. Ele está preocupado com a quantidade de cimento, com a profundidade do alicerce, com as pilastras de sustentação, analisa o terreno para saber se há algum risco, etc.

A diferença entre *design* e engenharia de *software* pode ser melhor compreendida com essa simples comparação. O *design* está interessado em tornar o ambiente do *software* agradável, de forma que o usuário se sinta bem ao utilizá-lo, não encontre grandes obstáculos e que os recursos estejam sempre a sua disposição de maneira clara e simples. Já os engenheiros de *softwares* estão preocupados com a estrutura interna do *software* e seus processos correlatos, como qual linguagem será usada, padrões de aplicativos, entre outros quesitos.

A correta aplicação dos princípios de *design* de interação somente poderão ser satisfatórios se correlacionados a outras áreas do conhecimento humano, tais como: Psicologia, Informática, Engenharia, Ciência da Computação, Fatores Humanos, Engenharia Cognitiva, Interação Homem Computador (IHC), Ergonomia Cognitiva, Sistemas de Informação e *Designers*. Pode parecer um pouco ampla essa concepção, porém,

o objetivo da aplicação de *design* é a elaboração de *softwares* que possam ser usados por qualquer pessoa.

Não há como desenvolver um produto bom se não se conhece muito bem as reais necessidades do usuário, como ele pensa e, na medida do possível, transportar para dentro do computador as habilidades que ele naturalmente já possui no mundo real. Dentro desse contexto é que a psicologia se torna uma ferramenta indispensável no processo de conhecimento do aprendizado humano e, portanto, deve ser aplicada dentro da Interação Humano Computador, no sentido de trazer conceitos e técnicas que possam facilitar o processo de aprendizagem e utilização do software.

O ser humano possui uma capacidade nata para entender, aprender e se adaptar; contudo, nos sistemas onde isso ocorra de forma natural, esse processo é bem mais simples. A palavra sistemas nesse contexto não deve ser somente entendida como um sistema computacional mas sim, numa aplicação mais ampla dentro dos sistemas humanos. Um bom exemplo disso é analisar o processo de aprendizagem de uma criança. Ela aprende por várias formas, sendo algumas dela: a comparação, a repetição, a intuição, a descoberta natural e gradativa do mundo. Em algum tempo, não muito longo, essa criança já está totalmente adaptada ao mundo e isso sem grandes traumas.

## PROCESSO DE APRENDIZADO

Talvez o cenário ideal para um sistema computacional fosse exatamente esse. Um sistema onde o usuário pudesse ir se adaptando gradativamente e de forma natural. Que o seu conhecimento prévio do mundo facilitasse o seu processo de aprendizado e adaptação no sistema informacional, onde ele fosse levado intuitivamente a descobrir novas coisas e se sentisse à vontade nesse processo.

Contudo, nem sempre esse trabalho interdisciplinar ocorre de forma simples. Exemplos podemos encontrar mesmo no entendimento dos problemas apresentados. Cada área do conhecimento possui uma visão própria sobre um determinado assunto e diversas soluções podem surgir para um mesmo problema, o que, até certo ponto pode ser até uma vantagem, mas dependendo da situação pode trazer sérios problemas de relacionamento entre os profissionais que formam a equipe. Existem alguns fatores que devem ser considerados no processo de *design* de interação, tais como:

- Identificar os requisitos do sistema a ser desenvolvido sendo essa atividade, básica e primordial em qualquer situação, além de constituir a fase inicial de qualquer processo. No livro *Design e avaliação de interfaces humano-computador*, as autoras Heloísa V. da Rocha e Maria C.C. Baranauskas (2000, p.31), fazem o seguinte comentário: “[...] conhecer o usuário é fundamental para se fazer o design de um sistema usável.”

- Com base nos requisitos acima levantados, elaborar *designs* alternativos de forma que satisfaçam completamente as especificações do sistema;

- Elaborar versões interativas dos *designs*, de forma que possam se comunicar e também que sejam analisadas.

- Avaliação constante durante todo o processo de desenvolvimento daquilo que está sendo produzido, pois é somente dessa forma que poderemos dizer que o produto é usável. Essa avaliação pode ocorrer de várias formas, como conversas com os usuários, testes da interface e desempenho e até mesmo questionários.

Tão importante quanto envolver os usuários nos testes de avaliação do *software* é saber identificar a sua rotina de trabalho no dia a dia. Muitas vezes é adequado reservar um tempo para ficar somente observando o usuário utilizar o sistema e com isso descobrir o que

pode ser modificado, o que pode ser readaptado e o que pode ser retirado do sistema.

A usabilidade de um *software* pode ser entendida como sendo o esforço necessário e realizado pelo usuário para conseguir utilizar o sistema e também como as técnicas usadas no sentido de facilitar a interação do usuário com o sistema através da interface do *software*.

Alguns indicadores podem dar uma boa visão sobre a usabilidade de um *software*. Um *software* pode ser classificado como sendo de boa usabilidade quando os seus usuários aprendem a utilizá-lo com facilidade (*learnability*), quando se lembram facilmente de tarefas anteriormente já realizadas (*memorability*), ou quando conseguem associar padrões entre as diversas telas de um mesmo sistema, através da ordem de disposição dos itens, rapidez no desenvolvimento de tarefas, o que leva à conclusão lógica de que a manipulação do *software* é bastante fácil e que por sua vez, essa facilidade eleve a produtividade do usuário (eficiência), e também, quando a taxa de erros durante a utilização do produto for baixa (erros).

O usuário de uma forma geral deve sentir-se bem ao utilizar o sistema, e esse é um fator extremamente importante (satisfação subjetiva). Um usuário satisfeito é um indicativo de que suas necessidades foram atendidas com a utilização do sistema.

Existe uma técnica para medir a usabilidade de um *software* denominada de “A Regra dos Dez Minutos”, PREECE Jennifer, ROGER Yvone, SHARP Helen (2005, p.37, *apud* Nelson, 1980). Através dessa regra, determina-se que um *software* é de boa usabilidade se um usuário inexperiente conseguir realizar alguma tarefa no *software* em menos de dez minutos, como o nome sugere.

Essa regra não pode ser aplicada a sistemas complexos, que envolvam uma

área muito específica do conhecimento, mas, via de regra, para sistemas comuns, é um bom termômetro e fornece pelo menos uma consideração inicial sobre possíveis problemas. Logo abaixo encontra-se uma citação que corrobora com o exposto acima:

Não se espera que um sistema computacional para arquitetos ensine arquitetura. Muito pelo contrário: a regra dos dez minutos exige que aquilo que um arquiteto já sabia seja útil no aprendizado do sistema. PREECE Jennifer, ROGERS Yvonne, SHARP Helen. (2005, p.37, *apud* Rubinstein e Hersch, 1984, p.9)

## PRINCÍPIOS DE DESIGN

Princípios de *design* são técnicas utilizadas para fornecer ao *Design* parâmetros para avaliar se seu produto é de boa usabilidade ou não. Existem alguns padrões de princípios, tais como: visibilidade, *feedback*, restrições, mapeamento, consistência e *affordance*.

**Visibilidade:** consiste em deixar disponível ao usuário um determinado objeto até que este seja necessário. Também é importante que haja um relacionamento entre o objeto e a ação que ele desempenha, pois isso facilita o entendimento de como os processos ocorrem. Exemplificando esse conceito, o botão de acendimento de um fogão a gás seria uma maneira simples de explicá-lo. Em qualquer fogão o botão tem normalmente a mesma localização e sempre executa a mesma tarefa, que é a de liberar o gás para o acendimento da chama.

**Feedback:** o *feedback* está também relacionado ao princípio da visibilidade e diz respeito ao retorno fornecido pelo sistema a uma determinada ação executada. Existem vários exemplos para essa situação, como por exemplo, o *click* do mouse, que ao ser pressionado produz o efeito sonoro do “*click*”. Com isso, temos certeza de que a ação foi efetuada. O alarme do carro que, quando pressio-

nado, faz com que o veículo dispare um sinal sonoro, indicando que realmente o dispositivo foi acionado.

Num sistema informatizado, um bom exemplo são as mensagens após um determinado comando, como uma mensagem de “Registro gravado com sucesso”, ou “Registro Excluído com Sucesso”. Isso fornece ao usuário uma sensação de segurança em face da ação desempenhada pelo sistema, ou seja, ele tem certeza de que foi realizado.

**Restrições:** estas podem ser entendidas como uma forma de inibir possíveis erros durante a realização de uma tarefa. Exemplos comuns de restrições são encontrados em menus de programas, quando uma tarefa não puder ser executada naquele momento, a opção que a executa estará inacessível ao usuário. Por exemplo, ao clicar no botão “inserir”, é comum que este seja automaticamente desabilitado, impedindo que o usuário clique novamente no mesmo local, podendo gerar um erro de execução do programa.

**Mapeamento:** o mapeamento está relacionado entre o controle e o efeito que este produzirá no mundo real. Exemplos de mapeamentos também são encontrados nos mais variados locais, sendo um deles o controle de um vídeo game. Neste existem setas de movimentação apontando para a direita, esquerda, para cima, para baixo e a sua ação no ambiente onde ele está inserido é intuitiva, ou seja, simplesmente ao ver uma seta apontando para cima, o usuário já deduz que, se pressionada, ela conduzirá a ação do jogo para cima, e assim sucessivamente.

**Consistência:** o conceito de consistência está relacionado à disposição coerente dos objetos na tela, procurando manter sempre um mesmo padrão na realização de operações semelhantes. Interfaces consistentes têm como algumas de suas vantagens, a facilidade no seu uso e aprendizagem.

Exemplificando a consistência de um programa, a utilização do botão direito do mouse pode ser citada. Em qualquer ícone em que ele seja utilizado, a última opção mostrada será sempre a que exibe as propriedades daquele objeto. Um outro exemplo são os programas desenvolvidos pela Microsoft, onde, em qualquer que seja o aplicativo, a disposição dos menus seguem um mesmo padrão. As funcionalidades do programa estão sempre agrupadas por categorias, que são exibidas num menu específico, tornando mais fácil o acesso a uma das centenas de operações que o aplicativo pode oferecer.

**Affordance:** refere-se à forma como um determinado objeto deverá ser utilizado através das suas propriedades, de forma intuitiva, ou seja, o simples fato de observar o objeto já fornece uma indicação de como se deve usá-lo. Exemplos: botões indicam que devem ser pressionados, é um conceito intuitivo, assim como alavancas sugerem que devam ser puxadas e volantes devem ser girados, entre inúmeros outros. Objetos com *affordance* dão ao usuário a certeza do que deve ser feito para utilizá-lo, sem a necessidade de instruções ou figuras.

## HEURÍSTICA

O termo *heurística* pode ser interpretado como sendo um conjunto de regras adquiridas com a própria experiência e a aplicação prática dos princípios de design. Indicam o que e como os princípios de design devem ser empregados.

O termo *Avaliação Heurística* é utilizado quando há uma inspeção de interface, realizada por um especialista, utilizando os princípios de usabilidade. A descrição dos princípios que se seguem foi baseada no livro *Design e avaliação de interfaces humano-computador*, as autoras Heloísa V. da Rocha e Maria C.C. Baranauskas (2000, p.48), a saber:

- **Visibilidade do status do sistema:** o sistema emite ao usuário um *feedback* do que está acontecendo, dentro de um tempo razoável.

- **Compatibilidade do sistema com o mundo real:** o sistema deve interagir com o usuário de uma forma simples, usando uma linguagem mais próxima do mundo real.

- **Controle do usuário e liberdade:** fornecer ao usuário “saídas de emergência” de lugares e/ou situações inesperadas ou inesperadas, como escolher uma função por engano ou errar uma operação.

- **Consistência e padrões:** manter um padrão no acesso às funções e na maneira como essas funções são descritas e indicadas, sem que o usuário fique em dúvida quanto a isso.

- **Prevenção de erros:** sempre que for possível, evitar que um erro ocorra.

- **Reconhecimento ao invés de memorização:** objetos, instruções, ações e opções devem estar visíveis ao usuário, sem que ele tenha que “guardar o caminho” toda a vez que quiser acessar uma opção do sistema.

- **Flexibilidade e eficiência no uso:** permitir aos usuários iniciantes a possibilidade de aumentar, com o tempo de uso, sua velocidade na interação com o sistema, ao mesmo tempo permitindo aos usuários experientes realizar tarefas com mais rapidez.

- **Estética e design minimalista:** evitar em diálogos o uso de informações irrelevantes ou raramente necessárias, pois elas concorrerem com as informações realmente importantes ao usuário.

- **Ajudar o usuário a reconhecer, diagnosticar e corrigir erros:** mostrar erros ao usuário em uma linguagem clara e simples de ser interpretada, sugerindo, se possível, uma maneira de solucionar o problema.

- **Help e documentação:** informações dessa natureza devem ser de fácil

localização, claras e objetivas, indicando soluções de maneira concreta.

Aplicando-se as normas e conhecimentos adquiridos no desenvolvimento do presente trabalho, foram utilizados para a elaboração do estudo de caso, do *Web Mail* fornecido pela CNEC Capivari, principalmente, os conceitos de princípios de *design* e a avaliação heurística proposta por Nielsen. Algumas deficiências foram constatadas, listadas e comentadas a seguir:

- 1) O usuário não escolhe o seu login. Este é gerado automaticamente pelo sistema, sendo este composto pelo seu primeiro nome mais o número do registro do aluno (Ex. andre2004d011@cneccapivari.br).

- 2) A tela inicial possui muitos componentes que não estão dispostos de maneira coerente, ferindo o princípio de consistência no *design* da Interface. O usuário tem dificuldade em localizar o que realmente necessita na tela.

- 3) Ao solicitar a exclusão de um item, o mesmo não é eliminado da tela e sim fica com um traço sobrescrito sobre o mesmo. A exclusão definitiva só acontece quando o usuário clica num *link* “eliminar excluídas” ou “esconder excluídas”. As mensagens excluídas ficam localizadas na caixa de entrada, não havendo distinção a não ser pelo traço sobrescrito.

- 4) Os nomes dos componentes não seguem o mesmo padrão de outros *Web Mails* usados pela maioria das pessoas. Por exemplo, a operação de criar novas mensagens, que na maioria dos *Web Mails* está no botão “Escrever” ou “Novo”, no *Web Mail* da CNEC está no botão “Compor”, sendo esse um termo de difícil compreensão principalmente para usuários iniciantes. A função sair do *Web Mail*, que na maioria está descrito simplesmente pelo botão “Sair”, está descrito como “Desconectar”, no *Web Mail* da CNEC.

- 5) O tópico de Ajuda, que tem como

objetivo ajudar o usuário a solucionar dúvidas de forma prática e simples, encontra-se em Inglês e praticamente não cumpre o propósito a que se propõe.

Citando novamente o livro *Design e avaliação de interfaces humano-computador*, as autoras Heloísa V. da Rocha e Maria C.C. Baranauskas (apud Nielsen, 2000, p.49), “Um dos principais conceitos em design que Nielsen dogmatizou, especialmente para o *design* de *websites*, foi a simplicidade”. Nielsen recomenda que se faça uma análise detalhada de cada componente da tela e que se retirem um a um esses componentes. Caso a funcionalidade não seja alterada pela retirada daquela componente, esse deve ser definitivamente excluído.

Dentro dessa linha conceitual, são encontradas no estudo de caso aqui apresentado, falhas graves que ferem principalmente o conceito de simplicidade, design minimalista, compatibilidade com o mundo real, e *help*, que na verdade não ajuda, pois as instruções são confusas e em outro idioma.

## CONCLUSÃO

A aplicação dos princípios de design e usabilidade são fatores indispensáveis para garantir uma interface mais

simplicista, propiciando ao usuário uma maior facilidade na sua utilização. São encontradas no mercado várias interfaces de difícil utilização, demonstrando com isso falhas de design e usabilidade. Essas falhas podem comprometer todo o programa. As funções que o programa executa podem não estar visíveis aos usuários de maneira clara, impossibilitando-as de serem adequadamente exploradas, embora o sistema se mostre eficiente, tecnicamente falando.

Fornecer ao usuário uma interface agradável e de fácil utilização é um passo importante na garantia da qualidade e aceitação do produto. Um programa intuitivo e de fácil utilização atende as necessidades do usuário e também aos requisitos de funcionalidade e usabilidade, requisitos esses que são essenciais para determinar a

qualidade de um *software*.

Conforme descrito no estudo de caso do *Web Mail* da CNEC Capivari, observou-se que alterações mínimas na interface garantiriam uma maior aceitação por parte dos usuários. Atualmente, a maioria dos alunos optam por outros *Web Mails* disponíveis no mercado por qualificá-los como sendo de mais fácil utilização, em detrimento ao *Web Mail* Institucional.

Aplicando-se os temas estudados nesse trabalho, conclui-se que é sempre conveniente optar por uma interface minimalista, intuitiva e que forneça ao usuário todos os recursos de que ele necessita para realizar seu trabalho. A observância de todos esses princípios não garantirá o sucesso do produto, mas por outro lado, reduzirá em muito as chances de o mesmo ser rejeitado. ■

andre@alleato.inf.br



## REFERÊNCIAS

PINHEIRO, Mauro; KLEMLZ, Laura. Métodos de pesquisa e desenvolvimento com foco no usuário na indústria de software: o caso do Globomail. Disponível em: <[http://www.feiramoderna.net/artigos/MauroPinheiro\\_LauraKlemz\\_SBDI2005.pdf](http://www.feiramoderna.net/artigos/MauroPinheiro_LauraKlemz_SBDI2005.pdf)>. Acesso em: 29 de agosto de 2006.

PREECE, Jennifer; ROGERS, Yvonne; SHARP,

Helen. Design de Interação: Além da interação homem-computador. Tradução Viviane Possamai. 1ª Edição. Porto Alegre: Editora Bookman, 2005.

ROCHA, Heloísa Vieira da; BARANAUSKAS, Maria Cecília Calani. Design e avaliação de interfaces humano-computador. São Paulo: IME-USP, 2000.