

NECESSIDADES PARA O DESENVOLVIMENTO DE UMA INTERFACE ADEQUADA PARA RESULTADOS DE ENSINO-APRENDIZAGEM BEM SUCEDIDOS.

TCC3047

FEVEREIRO /2006

José Antonio Gameiro Salles

UNISUAM / CCET / Desenv. de Softwares & UNISUAM/LAPEAD/PIBIC - antoniosalles@gmail.com

Carolina de Almeida Costa

UNISUAM / CCET / Desenv. de Softwares & UNISUAM/LAPEAD/PIBIC – karolmoura@gmail.com

Roberson Cotta Cardoso

UNISUAM / CCET / Ciên. da Computação – & UNISUAM/LAPEAD/PIBIC robsoncotta@gmail.com

Categoria: C – Métodos e Tecnologias

Setor Educacional: 3 - Educação Universitária

Natureza: A – Relatório de Pesquisa

RESUMO

Este trabalho demonstra o estudo sobre o projeto de criação de uma interface adequada para resultados de ensino-aprendizagem bem sucedidos, realizado durante o Projeto de Pesquisa da UNISUAM/PIBIC/LAPEAD. No decorrer deste trabalho estaremos analisando o que é uma interface, qual a sua importância em um projeto de software educativo, como uma interface bem estruturada e com layout bem definido e agradável interfere na qualidade da usabilidade da aplicação. Qual a metodologia educacional deve ser utilizada, observar com qual metodologia (conducionista ou construtivista) seu público alvo irá melhor se adaptar, qual o perfil do usuário, quais as características a serem observadas, e o processo de avaliação e testes de uma interface para que possa ser colocada em uso.

Palavras-chave: interface, ensino-aprendizagem.

1.INTRODUÇÃO :

Ao desenvolver um software educacional temos diferentes etapas no processo de criação tais como a programação, os testes e a criação de uma interface. Estaremos então analisando a fase de criação de uma interface e suas necessidades. Interface pode ser definida como a ligação entre o sistema e o usuário, o meio que permite o acesso ao sistema. Podemos ter interfaces simples, baseadas em textos mais utilizadas em sistemas antigos, ou interfaces gráficas que permitem a interação com o sistema não somente através do teclado, mas também do mouse ou outro dispositivo.

Ao definir uma interface estaremos tendo a responsabilidade de demonstrar através de ícones e menus todo o conteúdo do software, todas as suas funcionalidades, devendo assim otimizar todos os recursos disponíveis do sistema. Todavia não é aconselhável demonstrar todo seu conteúdo em uma única tela, visto que isso causaria um excesso de informação para o usuário o que poderia confundir-lo.

Ao criar uma interface devemos ter todo o cuidado em atrair o usuário para a aplicação, e mantê-lo na aplicação, pois qualquer ação repentina ou dificuldade que o usuário tenha pode fazer com que ele desista de utilizá-lo ou tenha que se submeter a um tempo maior em treinamentos.

2.ANÁLISE DO PERFIL DO USUÁRIO :

O perfil do usuário é o conjunto de informações, que descrevem as características relevantes do usuário no sistema, estudar o perfil do usuário é tarefa de grande importância na construção do software e sua interface.

São as características pessoais do usuário que influenciaram diretamente no desenvolvimento da interface do software. Usuários de diferentes faixas etárias, escolaridade e idade, por exemplo, provavelmente não possuirão as mesmas necessidades no que concerne a usabilidade do software. A análise de tarefas é responsável por determinar quais as tarefas o sistema deve realizar e como estas serão realizadas antes da implementação do software. Um ponto importante a ser considerado é a condição ambiental que o usuário trabalha, quais os objetivos dele ao realizar a tarefa, entre outros.

Projetar uma interface de fácil acesso ao usuário é tarefa de difícil complexibilidade para o programador, na construção de um software educacional ou não, o programador deve trabalhar juntamente a outros profissionais, como: animador(3D), designer, desenhista cadista e outros. Quando um programador decide construir um software algumas dificuldades podem aparecer, geralmente o programador não se preocupa com a interface e sim com a usabilidade do software, por isso é importante que o programador e sua equipe de trabalho testem suas aplicações com alguns usuários antes de colocarem a versão final do produto para uso.

3. ESCOLHA DA METODOLOGIA :

Ao desenvolver uma interface para ensino-aprendizagem devemos observar a necessidade de análise dos fundamentos das metodologias de ensino, bem como o público alvo para qual estaremos construindo a aplicação. Decidir qual a metodologia educacional utilizada fará com que comecemos a imaginar a aplicação pelo aspecto educacional, podendo ser utilizado o modelo conducionista de ensino, na qual será transmitido informações para o aluno, e logo após possivelmente ele será avaliado, caso não venha a assimilar o conteúdo aplicado, poderá retornar e tentar de novo.

Poderá também ser utilizado o modelo construtivista em que a interação entre tutor x aluno e aluno x aluno acontece com freqüência, tendo o aluno questões para resolver que irão estimulá-lo a buscar novos conhecimentos e aplicar os já adquiridos para responder as questões propostas.

Definir qual metodologia utilizar implicará em uma interface diferenciada para cada caso, o que incidirá diretamente na melhor utilização da aplicação. Uma abordagem pedagógica tradicionalista na maioria das vezes terá como foco uma tela com muita informação e questões para avaliação sobre o assunto abordado no formato de questões objetivas, já em uma interface voltada para a área construtivista poderá ter informações bem como espaços para interação entre aluno e tutor, além de formas de avaliação diferenciadas, visando observar se o aluno adquiriu novos conhecimentos tanto a matéria aplicada como se houve um aprendizado além do estabelecido pela aplicação.

4. A CONSTRUÇÃO DA INTERFACE:

Definido a metodologia utilizada surge o estudo da interface, que ira interferir em grande parte da qualidade da aplicação. Construir uma interface agradável e objetiva, de fácil manuseio e voltada para o perfil do usuário é de suma importância para o sucesso da aplicação. Uma vez que uma tela agressiva, ou com excesso de conteúdo ou muitos link's no menu podem confundir o usuário, deixando-o desmotivado a utilizar o sistema. Algumas pessoas que já tenham dificuldades em lidar com computadores podem não querer mais utiliza-lo senão entenderem que o mesmo pode auxilia-lo em inúmeras tarefas. Outros usuários até podem conseguir utilizar a aplicação, no entanto poderão cometer muitos erros, o que não seria interessante; em ambos os casos um treinamento prévio seria aconselhável.

Segundo Sommerville (2003, p.280) existem alguns princípios para o projeto de interface com o usuário, princípios estes que visam orientar o projetista na criação de um sistema dinâmico, e que atenda as exigências do cliente e do publico alvo, que faça com que o usuário tenha total aproveitamento de todas as funções, são eles: familiaridade com o usuário, consistência, mínimo de

surpresas, facilidade de recuperação, orientação do usuário, diversidade de usuários.

-Familiaridade com o usuário: ao projetar o sistema deve-se conhecer o público alvo, qual o tipo de usuário, para que possa se utilizar as mesmas formas de expressão que ele utiliza, ou seja, se você desenvolver uma aplicação para um biólogo, aplique os termos utilizados nesta área.

-Consistência: segue o sentido de manter os mesmos acessos para as mesmas funções, mesmo que você mude de tela, ou módulo. Utilizar os mesmos ícones para as mesmas funções é um exemplo, bem como teclas de atalho ou link's.

-Mínimo de surpresa: Muitas vezes por querer enfeitar o sistema ou chamar a atenção do usuário, o projetista incrementa o sistema de forma a surpreender, ou até mesmo assustar o usuário, o que não é aconselhável.

-Facilidade de recuperação: A interface deve possibilitar ao usuário recuperar erros que por ventura ele venha a praticar durante o uso do sistema. Funções de desfazer ou recuperar devem estar disponíveis para o usuário.

-Orientação do usuário: As opções de ajuda e notificações de erro devem ser explicativas ao usuário, uma mensagem de erro deve não só informar o erro como orientar o usuário na melhor maneira de resolve-lo.

-Diversidade de usuários: os recursos da interface devem atender aos diferentes tipos de usuários.

Observemos também que temos em geral dois tipos de usuários: os que conhecem o sistema e tem facilidade de lidar com ele, e os que pouco usam e por ventura esquecem o funcionamento correto do sistema. No primeiro caso a interface deve ser objetiva, pois o usuário interage com mais facilidade no sistema, no segundo caso o cliente precisara de telas de acesso mais fáceis e com recursos de ajuda.

As cores utilizadas em uma interface requerem uma atenção especial, Sommerville (2003,p.287) recomenda não mais que sete cores para um sistema , e que as mesmas devem ser utilizadas para mostrar a modificação de status de um sistema ou algum evento importante, ou seja, em uma aplicação de ensino, pudesse mudar as cores à medida que o usuário avança uma fase. Misturar cores tidas como “intensas” também pode ser prejudicial, visto que muitas pessoas tendem a confundi-las, e algumas das vezes essas cores como o vermelho e o azul juntas podem misturar-se e o usuário não distingui-las.

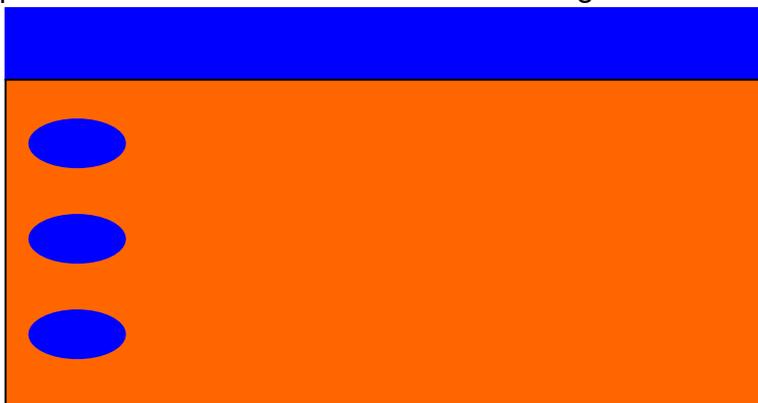


Figura 1: interface com cores intensas

5- ESCOLHA DO MELHOR PROTÓTIPO PARA IMPLEMENTAÇÃO :

Segundo (Pressman, 1995 p.399) um projeto de interface com o usuário é iterativo e para representa-lo pode ser usado um modelo espiral. A figura 2 mostra as tarefas de uma espiral. São elas:

1-análise e *modelagem* do usuário, tarefa e ambiente: verifica qual o tipo de usuário, qual o hardware ele irá utilizar, quais as tarefas ele pretende realizar com a aplicação.

2-Projeto de interface: nessa etapa estará a definição das ações da aplicação, bem como suas telas e funcionalidades.

3-Construção da interface (implementação): normalmente possui um protótipo utilizado para observar formas de utilização e cenários de uso. Conforme a aplicação for sendo construída ele pode ser implementada com o acréscimo de telas, ou mudanças de ícones e textos, de forma a aperfeiçoar o software.

4- Validação da interface: ocorre quando o usuário é capaz de utilizar a aplicação, aproveitando ao máximo suas funcionalidades.

Na fase de Implementação ou construção da interface pode ocorrer a prototipação, uma forma prática de observar o que foi projetado.

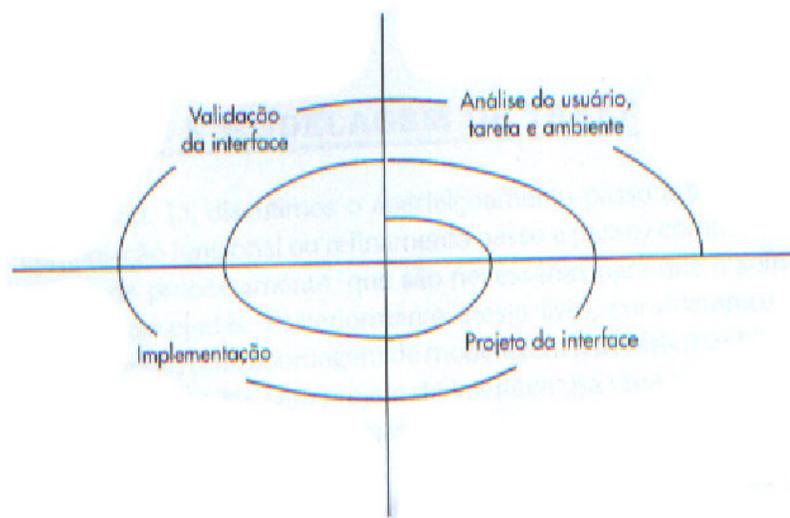


Figura 2: modelo espiral

6. AVALIAÇÃO:

Após a criação de um protótipo é ideal que ele seja testado e avaliado por alguns usuários, essa avaliação pode basear-se em apenas um teste ou até mesmo a questionários, em que poderá ser verificado o que deve ser corrigido, bem como as qualidades e defeitos do sistema. Sendo feita essa avaliação durante o processo de prototipagem poderá haver até mesmo uma economia de custos já que as mudanças poderão ser feitas antes de entrar em processo de produção final.

7. ALGUNS PONTOS IMPORTANTES PARA OBSERVAR :

- Pedir a confirmação, antes de apagar: faça com que a sua aplicação peça ao usuário, sempre, que confirme a decisão de apagar alguma coisa. Um aviso do tipo "você deseja realmente excluir", permitindo escolher entre sim ou não.

- Permitir que ícones sejam sempre ícones: quando os desenvolvedores insistem em ficar mudando o formato, a cor ou figura do ícone nas diferentes versões do aplicativo, ele acaba negando a principal finalidade do ícone, que é simplificar a vida do usuário. Escolha um ícone e permaneça com ele.

- Deverás indicar os campos obrigatórios: certificar-se de que o usuário saiba o que é obrigatório, numa tela. Você pode colocar uma pequena seta ou um asterisco, ao lado de cada um dos campos de preenchimento obrigatório.

8. CONSIDERAÇÕES FINAIS :

A interface é o elo de ligação entre o usuário e computador, em alguns casos é o primeiro contato do usuário com a máquina, se esse contato não for amigável, o usuário poderá desistir de usar. Criar uma interface amigável é uma tarefa complexa para a equipe de desenvolvimento, pois a partir da interface que será verificado a usabilidade do sistema, é nesse momento que o usuário decidirá se continuará a utilizá-lo ou não. E se utilizando conseguirá usufruir de todos os benefícios que a aplicação possui.

Nesse relatório de pesquisa observamos os aspectos relevantes para a estruturação de uma aplicação educacional que atenda em sua amplitude os fins para qual foi criada, adquirimos conhecimento para a criação de um projeto de software educacional, bem como para uma melhor análise dos já existentes.

9.REFERÊNCIAS BIBLIOGRÁFICAS:

-SOMMERVILLE, IAN; ENGENHARIA DE SOFTWARE. 6. ed. SAO PAULO: Addison Wesley, 2003.

-PRESSMAN, ROGER S.; ENGENHARIA DE SOFTWARE. SAO PAULO: Makron Books, 1995.

-NORTON, PETER; INTRODUÇÃO A INFORMATICA. SAO PAULO: Makron Books, 1996.