



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DANIEL PAULO DE ASSIS

**INTERFACE GRÁFICA COMO FATOR DETERMINANTE NO
DESENVOLVIMENTO DE SOFTWARE**

2012

Assis, SP



Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis - IMESA

DANIEL PAULO DE ASSIS

**INTERFACE GRÁFICA COMO FATOR DETERMINANTE NO
DESENVOLVIMENTO DE SOFTWARE**

**Trabalho de Conclusão de Curso (TCC) do
Instituto Municipal de Ensino Superior de
Assis - IMESA e a Fundação Educacional do
Município de Assis – FEMA.**

**Orientando: Daniel Paulo de Assis
Orientador: Prof. Dr. Luiz Ricardo Begosso**

2012

ASSIS, SP

FICHA CATALOGRÁFICA

PAULO DE ASSIS, Daniel

Interface Gráfica como fator determinante no Desenvolvimento de Software/ Daniel Paulo de Assis. Fundação Educacional do Município de Assis – Fema: Assis, 2012 92 p.

Orientador: Prof. Dr Luiz Ricardo Begosso

Trabalho de Conclusão de Curso (TCC) – Ciência da Computação - Instituto Municipal de Ensino Superior de Assis

1. GUI 2. Interface Gráfica. 3. IHC

CDD: 001.6
Biblioteca da FEMA

DANIEL PAULO DE ASSIS

**INTERFACE GRÁFICA COMO FATOR DETERMINANTE NO
DESENVOLVIMENTO DE SOFTWARE**

Trabalho de Conclusão de Curso (TCC) apresentado ao Instituto Municipal de Ensino Superior de Assis - IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito do Curso de Graduação, analisado pela seguinte comissão organizadora:

Orientando: Prof. Dr. Luiz Ricardo Begosso

Analisador (1): Prof. Me. Felipe Alexandre Cardoso Pazinato

2012

ASSIS, SP

DEDICATÓRIA

Dedico esse trabalho, à minha amada mãe Hosana Bispo de Assis, por ter lutado na vida para que eu pudesse concluir meus estudos, por estar presente sempre quando precisei, pelo colo, pelo carinho e pelo amor dispensado, ao meu pai Antonio Paulo de Assis (in memoriam) por ter sido a base de caráter e honestidade que me tornaram um homem melhor, à minhas irmãs Lídia e Telma, minha namorada Natália e aos meus poucos porém grandes amigos.

AGRADECIMENTO

Agradeço em primeiro lugar a Deus por ter me ouvido quando ninguém mais podia, por ter enxugado minhas lágrimas quando não quis compartilhá-las com mais ninguém, e por ter me dado forças pra seguir em frente e abençoado minha vida mesmo quando não fui merecedor.

À toda minha família, minhas irmãs, sobrinhos, meu amado pai e principalmente à minha amada mãe Hosana por ter sido meu porto seguro durante esse sonho, por sempre ter me apoiado nas escolhas que fiz até hoje e por ser meu grande exemplo de vida.

Aos meus amigos pessoais dentre os quais alguns posso citar nomes sem medo de errar, Renato (Júnior), Sérgio (Sérgião), Juliano (Gordo), Douglas (Dodô) vocês foram essenciais para essa conquista, pois me mantiveram forte e sempre trouxeram à minha vida muita felicidade.

À minha namorada Natália que chegou no último ano de faculdade, e não poderia ter chego em melhor hora, acalmou meu coração e me ajudou direta e indiretamente na conclusão desse trabalho.

À todos os professores que compartilharam seu conhecimento no decorrer do curso, em especial a professora Marisa por ter me levantado quando pensei em desistir, pelo seu apoio e incentivo que tornaram meu sonho possível.

Ao meu orientador Profº Luiz Ricardo Begosso, por ter aceito o desafio de me acompanhar nesse trabalho, pelo tempo, disposição e pela educação no trato com minha pessoa.

Aos meus amigos de curso, dentre os quais alguns se tornaram amigos pessoais, e que Deus permita essa amizade se estenda por toda a vida. Agradeço em especial ao meu grande amigo Pedro (in memorian) que nos deixou antes do combinado, mais que tornou os meus dias na faculdade e na vida mais divertidos e interessantes.

Enfim agradeço a todos aqueles que de alguma maneira contribuíram para a realização de um grande sonho, a conclusão de mais uma etapa em minha vida, é fato que hoje sou um homem mais realizado e conseqüentemente mais feliz.

EPÍGRAFE

“Quando você quer alguma coisa, todo o universo conspira para que você realize o seu desejo.”

Paulo Coelho.

RESUMO

Nos últimos tempos, o uso de sistemas computacionais aumentou consideravelmente, junto com esse crescimento a interatividade e a dinamicidade das aplicações cresceram também, o apelo visual sempre foi fator determinante para venda e usabilidade de produtos. Com o desenvolvimento de software não é diferente. Seja para pequenas ou grandes empresas ou até mesmo para desenvolvedores *free-lancer*, desenvolver uma boa interface gráfica, propor o desenvolvimento software como produto e um produto de qualidade e, proporcionar através de interfaces gráficas bem estruturadas um *feedback* maior com o usuário de sistemas é um requisito básico para conquistar um bom lugar no mercado de desenvolvimento de softwares. Este trabalho defende a ideia de que a interface gráfica é um dos fatores determinantes para o desenvolvimento de software, senão o mais importante deles, porque afinal de contas, para o usuário a Interface Gráfica é o sistema.

Palavras- chave: Interface Gráfica; Software; Desenvolvimento.

ABSTRACT

In the last times, the use of computer systems has increased considerably, and also with this growth, the interactivity and the dynamics of the applications grew too. The visual appeal has always been a determining factor for sales and usability of products, with the software development isn't different, for small or large businesses or even for free-lance developers, to develop a good graphical interface, to propose software development as a quality product and provide through well-designed graphical user interface, greater feedback to the user system is a basic to get good place in the market of software development. This work defends the idea that the graphical interface is one of the determining factors for the development of software, even the most important one, because after all, to the user, the graphical interface is the software.

Key words: Graphical interface; Software; Development.

Lista de Figuras

Figura 1 - Estrutura Hierárquica Áreas da Qualidade de Software (SWEBOK, 2004)..	22
Figura 2 Estrutura de Usabilidade	32
Figura 3 Funcionamento da Interface Usuário-Máquina (Rocha, 2000).	35
Figura 4 Simetria na Natureza.....	38
Figura 5 Simetria na Engenharia	38
Figura 6 Simetria na Arte (Homem Vitruviano de Da Vinci)	39
Figura 7 Snapshot Home Page Google	40
Figura 8 Percentual dos sentidos na captação das informações do meio	41
Figura 9 Snapshot Home Page CNN (Background na cor branca).....	46
Figura 10 Print Formulário Cadastro de Usuário (Rede Walmart).....	47
Figura 11 Ícone de exclusão (Cor vermelha).....	47
Figura 12 Ícone Salvar (Cor verde)	48
Figura 13 Logomarca Google e Microsoft (Mesmas cores na composição).....	48
Figura 14 Interação Usuário (touchscreen interação com ícones).....	49
Figura 15 Exemplos de ícones (Ajuda, Busca e Música).	50
Figura 16 Formulário cadastro de usuários (Americanas.com)	52
Figura 17 Snapshot Clientes e Fornecedores EngWhere	54
Figura 18 Menu Campus Sistema de Integração Factoring (http://campusinformatica.com.br/Help/Factoring/Hlp_foCadastroClientes.html)	55
Figura 19 Cadastro de Cliente Fidelit-Pharmanet (http://www.pharmanet.com.br/fidelity.htm)	56
Figura 20 Cadastro de Clientes Maxi Ação (http://www.maxiacao.com.br/si/site/0613)	57
Figura 21 Fechamento de venda Centertel (www.centertel.com.br).....	58
Figura 22 - Print InterfaceGraficaDPA na paleta do Netbeans	61
Figura 23 - Biblioteca InterfaceGraficaDPA adicionada as bibliotecas de um projeto (Netbeans).....	62
Figura 24 - InterfaceGraficaDPA adicionada a paleta da IDE Eclipse.....	63
Figura 25 - Biblioteca InterfaceGraficaDPA adicionada as bibliotecas de um projeto (Eclipse)	63
Figura 26 - Hierarquia de Classes Java.....	64
Figura 27 - Botões Personalizados.....	67
Figura 28 - Combobox Personalizados.....	68
Figura 29 - FormattedTextField Personalizados	68
Figura 30 - Labels (Ícones) personalizados	69
Figura 31 - Labels personalizados.....	69
Figura 32 - Panels personalizados	70
Figura 33 - Password personalizado.....	70
Figura 34 - TextArea personalizado	71
Figura 35 - TextField personalizado	71
Figura 36 - Menu Principal (InterfaceGraficaDPAPreview)	72
Figura 37 - Questão 1 Protótipo.....	73

Figura 38 - Questão 2 Agrupamento.....	74
Figura 39 - Questão 3 Simetria	74
Figura 40 - Questão 4 Padrão entre todas Interfaces	75
Figura 41 - Questão 5 Cor (Background)	75
Figura 42 - Questão 6 Fonte (Tamanho Mínimo).....	76
Figura 43 - Questão 7 Ícones (Figuras).....	76
Figura 44 - Questão 8 Ícones (Legenda).....	77
Figura 45 - Questão 9 Mensagens de alerta	77
Figura 46 - Questão 10 Help (Ajuda)	78
Figura 47 - Resultado de um teste com 100 pontos.....	78
Figura 48 - Fases do Processo de Prototipação	79
Figura 49 - Protótipo Interface Cadastro Dados Pessoais e Endereço	81
Figura 50 - Interface de exemplo Implementada com base em protótipo	82
Figura 51 - Protótipo de uma tela de Login	83
Figura 52 - Interface de exemplo (Login) implementada com base em protótipo.....	83
Figura 53 - Protótipo de tela para cadastro de usuários de sistema.....	84
Figura 54 - Interface de exemplo(Cadastro de Usuários) implementada com base em protótipo	85
Figura 55 - Protótipo de Menu utilizando somente ícones com show hint.....	85
Figura 56 - Interface de Exemplo (Menu) implementada com base em protótipo	86
Figura 57 - Interface de Exemplo (Ícones com dicas)	87
Figura 58 - WebSite InterfaceGraficaDPA.....	88

Lista de Tabelas

Tabela 1 Características da Tarefa (Aplicação x Exemplos).....	27
Tabela 2 Auto Descrição (Aplicação x Exemplos)	28
Tabela 3 Controlabilidade (Aplicação x Exemplos)	28
Tabela 4 Conformidade com expectativas do usuário (Aplicação x Exemplos).....	29
Tabela 5 Tolerância ao erro (Aplicação x Exemplos)	29
Tabela 6 Adequação a Individualização (Aplicação x Exemplos)	30
Tabela 7 Adequação ao aprendizado (Aplicação x Exemplos).....	30
Tabela 8 Exemplo medidas de usabilidade.....	34

Sumário

1. INTRODUÇÃO	15
1.1 OBJETIVOS	16
1.2 MOTIVAÇÃO	16
1.3 JUSTIFICATIVA.....	17
1.4 ESTRUTURA DO TRABALHO.....	18
2. FUNDAMENTAÇÃO TEÓRICA.....	19
2.1. ENGENHARIA DE SOFTWARE	19
2.2. USABILIDADE	20
2.3. QUALIDADE DE SOFTWARE	21
2.4. GUI (GRAPHICAL USE INTERFACE).....	23
2.5. IHC (INTERAÇÃO HUMANO-COMPUTADOR).....	23
3. ESTUDO DE CASO	25
3.2. NORMA NBR ISO 9241 PARTE 10 PRINCÍPIOS DE DIÁLOGO	25
3.2.1. Princípios de Diálogo	26
3.2.2. Características do Usuário	26
3.2.3. Características da Tarefa	26
3.2.4. Auto Descrição.....	27
3.2.5. Controlabilidade	28
3.2.6. Conformidade com expectativas do usuário	28
3.2.7. Tolerância ao erro	29
3.2.8. Adequação a Individualização.....	29
3.2.9. Adequação ao aprendizado	30
3.3. NORMA NBR ISO/IEC 9421-11 PARTE 11 ORIENTAÇÕES SOBRE USABILIDADE	30
3.3.1. Especificando e medindo a usabilidade de produtos(NBR ISO 9421-11)32	
3.3.2. Desenvolvimento de produto	33
3.3.3. Exemplo de medidas de usabilidade.....	33
3.4. A INTERFACE GRÁFICA.....	34
3.5. OTIMIZANDO A INTERFACE	36
3.5.1. Simetria	37
3.5.2. Cor	40
3.5.3. Ícones.....	49

3.5.4.	Fontes.....	50
3.5.5.	Layout.....	51
3.6.	ESTUDOS DE CASO (INTERFACES FALHAS EXISTENTES NO MERCADO).....	53
4	Desenvolvimento	58
4.1	JAVA.....	59
4.2	DESCRIÇÃO (INTERFACEGRAFICADPA).....	60
4.3	INTERFACEGRAFICADPA (BIBLIOTECA SWING).....	61
4.3.1	InterfaceGraficaDPA (Conceitos Java).....	64
4.3.2	InterfaceGraficaDPA (Componentes).....	65
4.4	INTERFACEGRAFICADPA (APLICATIVO).....	72
4.5	INTERFACEGRAFICADPA (QUESTIONÁRIO).....	73
4.6	PROTOTIPAÇÃO	79
4.6.1	Prototipação e Formulários de Exemplos.....	80
4.6.2	Formulário (Dicas).....	86
5	Conclusão.....	87
5.1	CONSIDERAÇÕES FINAIS.....	87
5.2	TRABALHOS FUTUROS	89
	REFERÊNCIAS	90

1. INTRODUÇÃO

Ergonomia, usabilidade e qualidade de software. Todas essas áreas da Engenharia de Software, são fundamentais para que um software (aplicativo, utilitário, instalador) possua um ciclo de vida regular e não caia em desuso, não cause repulsa por parte dos usuários e o seu uso proporcione o mínimo de conforto.

Sem dúvida essas áreas demonstraram estar em grande crescimento, visto que em uma relação custo X benefício os lucros saem favorecidos. Desde o lançamento pela Microsoft do Windows 1.0 em 20 de Novembro de 1985 (Microsoft, 2012) a necessidade de trazer um ambiente cada vez mais agradável, prático e intuitivo ao usuário vem crescendo. Na verdade o S.O. Windows surgiu justamente com esse objetivo: substituir o MSDOS (Microsoft Disk Operating System) que devido a sua dificuldade de utilização, limitava drasticamente a quantidade de usuários e conseqüentemente suas vendas.

Assim, a cada dia que passa, a parcela da população que utiliza softwares nas tarefas do cotidiano aumenta gradativamente e, conseqüentemente, a produção de softwares também, mas nem sempre acompanhada da qualidade devida. Deste modo, a interface dos sistemas é a parte que fornece ao usuário a possibilidade de utilizar um computador, celular, palmtop, sem a necessidade de ser um “*expert*” no assunto, fato este que nem sempre acontece.

Desta forma, desenvolver interfaces que tornem o uso da máquina transparente, que forneça o máximo de conforto e segurança ao usuário é de fundamental importância para a sobrevivência de um software. É com este objetivo, o de facilitar as relações entre usuários e sistemas computacionais através de melhorias nas interfaces dos Softwares, que esse trabalho se norteia.

1.1 OBJETIVOS

Esse trabalho tem como objetivo principal, estudar a fundo metodologias que possibilitem o desenvolvimento de softwares que proporcionem uma maior facilidade no seu uso e que também minimizem o estresse e o cansaço provocado por ações rotineiras, complexas ou que demandem grande responsabilidade.

Assim, com base nos conhecimentos obtidos, objetiva-se também a construção de Interfaces mais “amigáveis”, intuitivas e que facilitem a vida do usuário de sistemas computacionais, e que também possam servir de base para estudos futuros.

Através desses estudos, visa-se construir um material rico em teoria e prática de boas técnicas para construção de um software, que possa unir de maneira concisa a usabilidade e a qualidade, e que respeite as normas de ergonomia existentes.

Por fim, dentre os objetivos almejados, também se espera resolver a carência de materiais do tipo e solucionar um grande problema que é a inutilização de um software após sua implantação devido à complexidade de uso, dificuldade de entendimento, cores agressivas e outros fatores que possam vir a podar seu ciclo de vida.

1.2 MOTIVAÇÃO

De acordo com NBR ISO/IEC 9126-1 (2003), os computadores têm sido usados numa variedade de áreas de aplicação cada vez maior e sua operação correta é frequentemente crítica para o sucesso de negócios e para a segurança humana. Deste modo, desenvolver ou selecionar produtos de software de alta qualidade é de importância primordial.

A aplicação das técnicas e metodologias que serão estudadas nesse trabalho, certamente proporcionarão aos profissionais e empresas da área de desenvolvimento grandes benefícios como, por exemplo:

- satisfação do cliente;
- menor custo com setor de Help Desk;
- menor custo com treinamento de usuário;
- maior aceitação do software pelos usuários;
- evitar a inutilização do Software;
- aumento de vendas;
- retenção de consumidores;
- evitar o fechamento prematuro de pequenas empresas de Software.

Como já mencionado, as pesquisas, materiais estudados e publicados em tais assuntos são notoriamente escassos, deste modo, a necessidade de auxílio motiva a construção de um material de apoio e suporte aos que necessitem, onde, além dos benefícios proporcionados aos profissionais e empresas da área, o maior beneficiado em questão será o usuário final, que poderá fazer uso de um software mais “amigável”.

A ideia é que também ocorra o surgimento de uma nova mentalidade entre os profissionais da área, onde o usuário final possa interagir da maneira mais transparente possível, evitando o clichê “É feio, mais funciona”.

1.3 JUSTIFICATIVA

Esse trabalho justifica-se, em primeiro lugar, pela grande carência de materiais sobre o assunto (especificamente em Português). Também visa trazer a conhecimento público algumas instituições e normas que regem o uso de trabalho com computadores. Não menos importante é a justificativa do crescimento de softwares natimortos, por se enquadrarem em modelos de desenvolvimento onde ergonomia, usabilidade e qualidade de software não

foram priorizadas. Visa também à diminuição da construção de softwares em que a aparência se torna o maior empecilho para sua utilização.

Assim, a partir da afirmação de Hyman (1994), “sistemas complexos com falhas no projeto das interfaces e na interação homem-sistema podem contribuir para as falhas humanas”, este trabalho objetiva-se pela ideia de, através da melhoria nas interfaces, diminuir o número de falhas operacionais nos sistemas de computadores.

1.4 ESTRUTURA DO TRABALHO

Para um melhor entendimento e organização este trabalho foi dividido em capítulos que serão especificados a seguir.

O primeiro capítulo apresenta o contexto, os objetivos, a justificativa e a motivação para o desenvolvimento do projeto.

No segundo capítulo tem-se o embasamento teórico de fundamentos básicos da Engenharia de Software.

No terceiro capítulo foram feitos estudos de caso, abordando normas, metodologias e conceitos para o desenvolvimento de uma boa interface gráfica.

Seguindo o cronograma estipulado serão desenvolvidos ainda o quarto capítulo com a modelagem e a prototipação das interfaces à serem desenvolvidas, um quinto capítulo com o desenvolvimento das interfaces e o sexto e último capítulo com as conclusões obtidas com o encerramento do trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados conceitos que são essenciais para o desenvolvimento desse trabalho, a princípio um breve resumo sobre Engenharia de Software, seguido de usabilidade e qualidade de software. Também serão abordados os conceitos referentes à GUI (Graphical User Interface) e IHC (Interação Humano- Computador).

2.1. ENGENHARIA DE SOFTWARE

Segundo Naur (1969), Engenharia de Software é o estabelecimento e o uso de princípios sólidos de engenharia para que se possa obter economicamente um software que seja confiável e que funcione eficientemente em máquinas reais.

Desta forma, entende-se que Engenharia de Software é a "aplicação de uma abordagem sistemática, disciplinada e quantificável no desenvolvimento, na operação e na manutenção de software, ou seja, a aplicação de engenharia de software, além do estudo de tais abordagens" (IEEE, 1990).

A Engenharia de Software trata-se também de uma disciplina da engenharia que se dedica exclusivamente aos estudos dos processos que envolvem a produção de um software.

Segundo SWEBOK (2004), as origens do Corpo de conhecimento da Engenharia de Software são as seguintes:

- Matemática
- Ciência da Computação
- Administração de Projetos
- Ciências Cognitivas e Fatores Humanos
- Engenharia de Computadores
- Administração e Ciências
- Administrativas

- Engenharia de Sistemas

A Engenharia de Software é dividida no SWEBOK (2004), em um total de onze áreas de conhecimento (KA: Knowledge Area): requisitos, gerência de engenharia, projeto, métodos e ferramentas de engenharia, construção, processo de engenharia, testes, qualidade, manutenção, disciplinas relacionadas e gerência de configuração.

Resumindo, a “Engenharia de Software se dedica aos problemas práticos da produção de software” (Sommerville, 2006).

2.2. USABILIDADE

Jordan (1998) define de maneira informal o termo Usabilidade como o quanto é fácil utilizar determinado produto. Já a ISO NBR 9241-11 (2000), afirma que Usabilidade “é a eficácia, a eficiência e a satisfação com que usuários específicos conseguem alcançar objetivos específicos em ambientes particulares”. E a ISO/IEC 9126-1 (2003), define Usabilidade como “a capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas”.

Neste sentido, Nielsen (1994) um dos maiores especialistas em usabilidade, propõe um conjunto de dez heurísticas de usabilidade:

- visibilidade do estado do sistema;
- mapeamento entre o sistema e o mundo real;
- liberdade e controle ao usuário;
- consistência e padrões;
- prevenção de erros;
- reconhecer em vez de lembrar;
- flexibilidade e eficiência de uso;
- design estético e minimalista;
- suporte para o usuário reconhecer, diagnosticar e recuperar erros;
- ajuda e documentação.

Assim, pode-se entender que a usabilidade tem relação direta com o usuário, afinal é ele quem determina se um produto, neste caso o software, trará mais produtividade e otimizará as tarefas do dia-a-dia. Desta forma, a usabilidade é foco no usuário, portanto para a construção de um software com um padrão aceitável de usabilidade é necessário que se tenha um bom conhecimento dos usuários finais.

2.3. QUALIDADE DE SOFTWARE

Segundo o Aurélio (2004), Qualidade é superioridade, excelência de alguém ou de algo, tal definição serve também para o desenvolvimento de software, porém, segundo SWEBOK (2004), qualidade de software é o conjunto de atividades relacionadas com garantia de qualidade de software, entre estas as atividades de verificação e validação.

Em meados dos anos 1940, surgiram várias organizações que tratavam especificamente de qualidade, entre elas pode-se citar algumas mais conhecidas e que se mantêm até os dias hoje, servindo como base para projetos em diferentes áreas:

- ASQC (American Society for Quality Control)
- ABNT (Associação Brasileira de Normas Técnicas)
- ISO (International Standardization Organization)

A Figura 1 a seguir, mostra, segundo o SWEBOK (2004), a estrutura hierárquica que subdivide as áreas da Qualidade de Software.

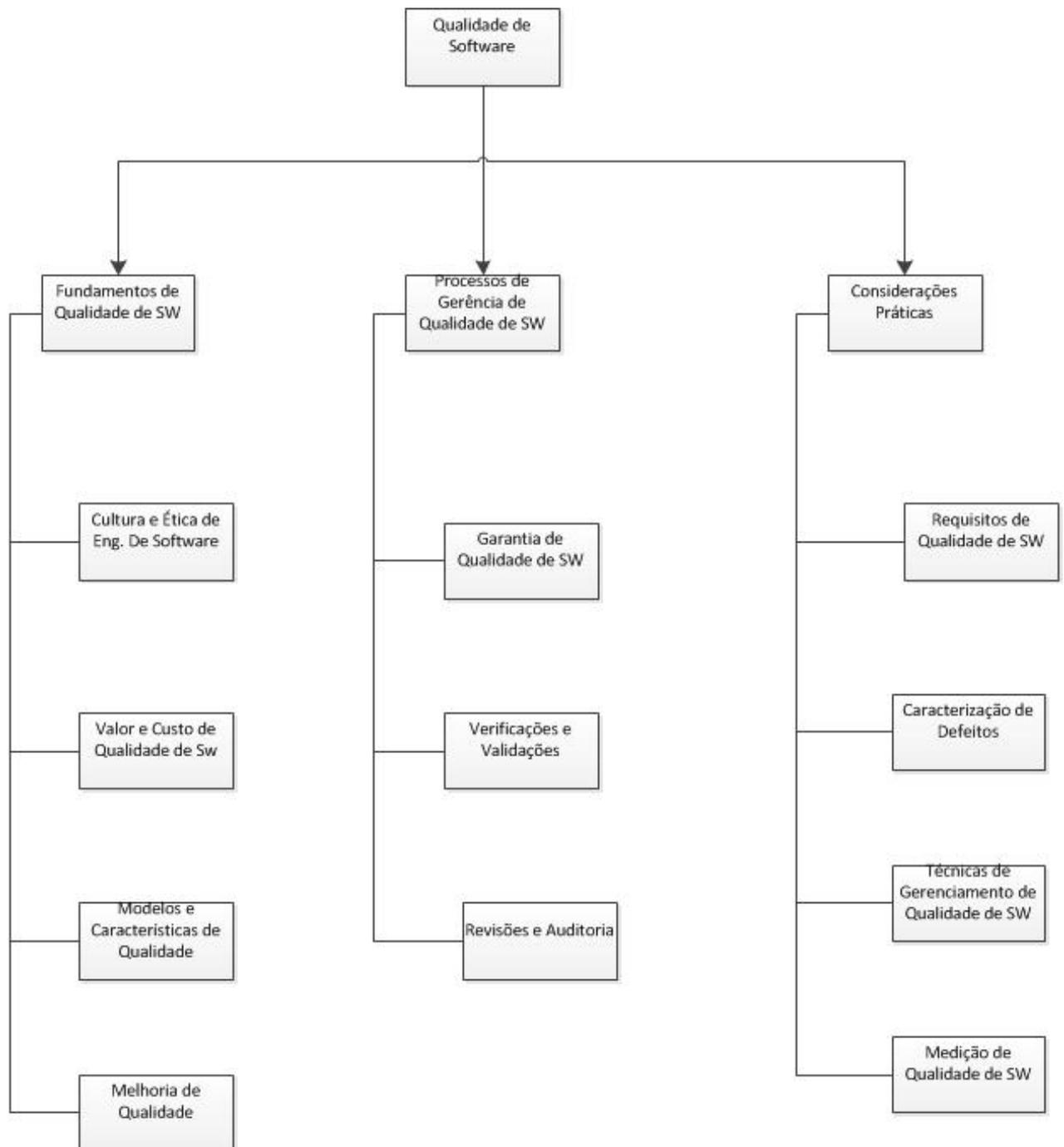


Figura 1 - Estrutura Hierárquica Áreas da Qualidade de Software (SWEBOK, 2004).

2.4. GUI (GRAPHICAL USE INTERFACE)

Para Beltrán (2002), “uma interface inspira sentimentos de rejeição e aceitação que determinam a maneira como o usuário irá se comportar diante de um sistema computacional”. Segundo o mesmo autor, “86% das pessoas que decidem abandonar o uso de um programa, o fazem devido à interface”.

“Qualquer que seja o meio de comunicação: textual, visual, audiovisual ou eletrônico, há sempre uma interface que media a interação, seja o layout de um livro, o aparelho de telefone, o controle e a tela de televisão, o teclado e o monitor do computador, para citar alguns exemplos. Assim, cada tipo busca com seu design próprio, tornar-se de fácil uso e mais atraente possível para o usuário.” (Silva, 1998).

A “Graphical User Interface foi desenvolvida em 1981 pela Xerox PARC por Alan Kay, Douglas Engelbart, e um grupo de outros pesquisadores, em seguida a Apple lançou o computador Lisa, primeiro computador disponível comercialmente em 19 de janeiro de 1983.” (GUI, Computer Hope, 2012).

A GUI usa janelas e ícones e menus para executar comandos, é uma das partes mais importantes do Software, pois, através dela o usuário interage com a aplicação e obtém os resultados esperados. Nos dias atuais, as IDE's para desenvolvimento, fornecem inúmeros recursos e componentes (botões, textfields, labels) que facilitam a vida do desenvolvedor de Interfaces.

2.5. IHC (INTERAÇÃO HUMANO-COMPUTADOR)

IHC “é a área que trata exclusivamente do relacionamento dos seres humanos com as máquinas (computadores). As primeiras referências ao termo IHC começaram no final da década de 1970” (Brandão, 2006); academicamente, a IHC “ganhou impulso nos anos 90 com o surgimento de eventos específicos sobre o assunto que incentivaram a publicação de pesquisas e reflexões” (Endler, Pimenta, 2004).

A IHC “tem um importante papel no projeto e no desenvolvimento de todos os tipos de sistemas computadorizados, como controle de tráfego aéreo, processamento nuclear, aplicativos de escritório ou jogos” (Brandão, 2006).

Segundo NORMAN (2003), “não é suficiente projetar um software que apenas funcione, ou que apenas possua funcionalidades necessárias para o cumprimento da tarefa”, ou seja, não se pode deixar a cargo do usuário a responsabilidade de manipular ou lidar com as complexidades de um sistema, daí a necessidade da implantação das metodologias de IHC.

3. ESTUDO DE CASO

Neste capítulo serão realizados estudos sobre as normas NBR ISO 9241 Parte 10, NBR ISO/IEC 9421-11 Parte 11 que regem os requisitos ergonômicos para trabalhos com computadores.

Também nesse tópico serão abordados os erros mais comuns no desenvolvimento de software no que se refere ao assunto principal desse trabalho, as interfaces gráficas, através de avaliações de softwares existentes no mercado.

Durante o decorrer desse capítulo serão apresentados de maneira resumida tópicos e exemplos mais relevantes que constituem as normas citadas acima mantendo a máxima fidelidade com o conteúdo original pra que não ocorra a descaracterização das normas.

3.2. NORMA NBR ISO 9241 PARTE 10 PRINCÍPIOS DE DIÁLOGO

A NBR 9241 fornece princípios ergonômicos formulados em termos gerais, isto é, eles são apresentados sem referência às situações de uso, aplicação, ambiente ou tecnologia. Pretende-se que estes princípios sejam usados na especificação, projeto e avaliação de diálogos para computadores. (NBR ISO 9241 Parte 10, 2000).

Esta décima parte da Norma NBR ISO 9241-Requisitos ergonômicos para trabalho de escritório com computadores, trata especificamente de Princípios de diálogo, onde diálogo é definido como a interação entre um usuário e um sistema para alcançar um objetivo particular. Vale ressaltar que apesar da pouca probabilidade do usuário final ter conhecimento da norma ele será o maior beneficiário dos projetos que utilizarem a mesma.

3.2.1. Princípios de Diálogo

De acordo com a norma, foram definidos sete princípios de grande importância para o projeto e também para a avaliação de um diálogo com computadores, os princípios são listados a seguir:

- adequação à tarefa;
- auto descrição;
- controlabilidade;
- conformidade com expectativas do usuário;
- tolerância ao erro;
- adequação à individualização;
- adequação ao aprendizado.

3.2.2. Características do Usuário

Os princípios de diálogo devem se fundamentar nas características do usuário para sua aplicação, algumas das características do usuário sugeridas pela norma são:

- capacidade de atenção;
- limites da memória de curto termo;
- comportamento de aprendizado;
- nível de experiência no trabalho e no sistema;
- visão internalizada da estrutura de base, e da proposta do sistema com o qual o usuário irá interagir.

3.2.3. Características da Tarefa

O desempenho em uma tarefa do sistema está relacionado diretamente com as características dos diálogos do sistema, podendo variar obviamente com as necessidades de cada sistema.

Segundo a NBR ISO 9241 parte 10 (2000) um diálogo é adequado para a tarefa quando ele apoia o usuário em uma conclusão efetiva e eficiente da tarefa.

A tabela 1 descreve alguns exemplos citados pela NBR ISO 9241 em uma relação Aplicação x Exemplos.

Aplicação	Exemplos incluem
Recomenda-se que o diálogo apresente ao usuário somente informações relacionadas com a conclusão da tarefa.	A formatação da informação tal como cor, e informações tais como dia atual, data, etc. são apresentadas somente se elas facilitarem a conclusão da tarefa.
Convém que qualquer ação que possa ser apropriadamente alocada ao software de interface para execução automática deva ser realizada pelo software sem o envolvimento do usuário.	O cursor é automaticamente posicionado no primeiro campo de entrada relevante para a tarefa. Os procedimentos de inicialização do sistema são realizados automaticamente.
Se existirem recursos de entradas pré-definidos para uma dada tarefa (p.ex. valores pré-definidos como padrão), convém que não seja necessário que o usuário tenha que entrar tais valores. Convém também que seja possível substituir valores pré-definidos por outros valores ou por outros valores pré-definidos apropriados.	Se a data atual é requerida pela tarefa, não é necessário que ela seja digitada, mas o usuário é capaz de modificá-la.

Tabela 1 Características da Tarefa (Aplicação x Exemplos)

3.2.4. Auto Descrição

Um diálogo é auto descrito quando cada passo do diálogo é imediatamente compreensível por meio de resposta do sistema ou é explicado, sob demanda, ao usuário. A tabela 2 descreve os exemplos.

Aplicação	Exemplos incluem
Após qualquer ação do usuário, convém que o diálogo apresente resposta do sistema onde apropriado. Se a ação do usuário pode trazer consequências severas, convém que o sistema proporcione explicação e solicite confirmação antes de executar a ação.	É necessário repercutir a atividade do teclado junto com modificação do estado de dados para ajudar o usuário a compreender o que aconteceu na aplicação e o que o usuário pode controlar. A aplicação indica se o diálogo pode ser revertido fornecendo informação explícita sobre o que pode ser revertido.

	O sistema de diálogo pede confirmação se uma exclusão não puder ser revertida.
Quando uma entrada é solicitada, convém que o sistema forneça informações ao usuário sobre a entrada esperada.	O sistema de diálogo solicita a entrada através da apresentação do nome do campo juntamente com informações sobre o tipo de dado (p.ex. data, números, bandeira) e o formato de entrada (p.ex. dd.mm.aaaa)
Convém que as mensagens sejam formuladas e apresentadas em um estilo compreensivo, objetivo e construtivo, dentro de uma estrutura consistente. Convém que as mensagens não contenham nenhum julgamento de valor, tal como “Esta entrada não é válida”.	Mensagem: “Para data de aniversário, por favor, use o seguinte formato: DD/MM/AAAA”.

Tabela 2 Auto Descrição (Aplicação x Exemplos)

3.2.5. Controlabilidade

Um diálogo é controlável quando o usuário pode iniciar e controlar a direção e o ritmo da interação até que o objetivo tenha sido atingido. A tabela 3 mostra os exemplos.

Aplicação	Exemplos incluem
Convém que os sistemas de diálogos forneçam ao usuário controle sobre como continuar com o diálogo.	O sistema de diálogo posiciona o cursor no próximo campo de entrada, mas oferece ao usuário a possibilidade de selecionar um campo diferente.
Convém que a velocidade de interação não seja imposta pela operação do sistema. Convém que a mesma esteja sempre sob controle do usuário de acordo com suas necessidades e características.	Nenhum campo de entrada é apagado substituído ou tornado indisponível ao usuário até que o mesmo sinalize haver completado a entrada de dados, por exemplo, pressionando a tecla [Enter].

Tabela 3 Controlabilidade (Aplicação x Exemplos)

3.2.6. Conformidade com expectativas do usuário

Um diálogo está em conformidade com as expectativas do usuário quando é consistente e corresponde por um lado, às características do usuário, tais como

conhecimento da tarefa, educação e experiência, e por outro lado, às convenções usualmente aceitas. A tabela 4 fornece os exemplos.

Aplicação	Exemplos incluem
Convém que o comportamento e a aparência da interface de um sistema de diálogo sejam consistentes.	Mensagens referentes ao estado do sistema sempre aparecem na mesma linha. A mesma tecla é sempre usada para finalizar o diálogo.
Convém que o cursor esteja onde a entrada é solicitada.	O sistema de diálogo move o cursor automaticamente para a próxima posição onde o usuário espera fazer a próxima entrada.

Tabela 4 Conformidade com expectativas do usuário (Aplicação x Exemplos)

3.2.7. Tolerância ao erro

Um diálogo é tolerante ao erro se, apesar de erros de entrada evidentes, o resultado esperado pode ser obtido com pouca ou nenhuma ação corretiva do usuário. A tabela 5 descreve os exemplos.

Aplicação	Exemplos incluem
Convém que os erros sejam explicados para ajudar o usuário a corrigi-los.	Na medida em que o sistema seja capaz de identificar os erros, ele apresenta uma mensagem contendo informações sobre a ocorrência do erro, tipo do erro e possíveis métodos de correção.
Convém que a validação/verificação de dados aconteça antes do processamento da entrada. Convém que controles adicionais sejam fornecidos para comandos com consequências sérias.	Uma lista de itens a serem apagados é apresentada antecipadamente. A exclusão efetiva acontece somente sob confirmação.

Tabela 5 Tolerância ao erro (Aplicação x Exemplos)

3.2.8. Adequação a Individualização

Um diálogo é capaz de individualização quando o software de interface pode ser modificado para se adequar às necessidades da tarefa, preferências individuais e habilidades do usuário. A tabela 6 descreve os exemplos.

Aplicação	Exemplos incluem
Convém que o usuário possa estabelecer parâmetros operacionais de tempo adequados às suas necessidades individuais.	O sistema de diálogo permite ao usuário controlar a velocidade de rolagem da apresentação das informações.
Convém que os usuários possam escolher entre diferentes técnicas de diálogo para diferentes tarefas.	O sistema de diálogo permite ao usuário iniciar um diálogo tanto digitando um comando quanto selecionando uma opção de menu.

Tabela 6 Adequação a Individualização (Aplicação x Exemplos)

3.2.9. Adequação ao aprendizado

Um diálogo é adequado quando apoia e guia o usuário no aprendizado para usar o sistema. A tabela 7 descreve os exemplos.

Aplicação	Exemplos incluem
Convém fornecer diferentes formas para ajudar o usuário a se familiarizar com os elementos de diálogo.	Lugares padronizados são usados para o mesmo tipo de mensagem. Distribuições similares de elementos de tela são usadas para tarefas com objetivos similares.

Tabela 7 Adequação ao aprendizado (Aplicação x Exemplos)

3.3. NORMA NBR ISO/IEC 9421-11 PARTE 11 ORIENTAÇÕES SOBRE USABILIDADE

Segundo a NBR ISO/IEC 9421-11 (2000) usabilidade é a medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso.

Essa norma trata especificamente do conceito de usabilidade, observando o software como um produto ela se enquadra perfeitamente para ajudar a lapidar os objetivos desse projeto.

Usabilidade é uma consideração importante no projeto de produtos uma vez que ela se refere à medida da capacidade dos usuários em trabalhar de modo eficaz, efetivo e com satisfação. (NBR ISO 9421-11, 2000).

A norma também enfatiza que a usabilidade dos produtos pode ser melhorada pela incorporação de características e atributos conhecidos como capazes de beneficiar os usuários em um contexto particular de uso. O nível de usabilidade é medido através do desempenho e satisfação dos usuários, neste caso específico sistemas para computadores, porém a NBR ISO 9421-11 parte 11 não cobre os processos de desenvolvimento de sistema.

A abordagem adotada na NBR 9241-11 (2000) tem benefícios que incluem:

- A estrutura pode ser usada para identificar os aspectos de usabilidade e os componentes do contexto de uso a serem considerados no momento da especificação, projeto ou avaliação de usabilidade de um produto.
- O desempenho (eficácia e eficiência) e a satisfação dos usuários podem ser usados para medir o grau em que um produto é usável em um contexto particular.
- Medidas de desempenho e satisfação dos usuários podem fornecer uma base de comparação da usabilidade relativa de produtos, com diferentes características técnicas, que são usados no mesmo contexto.
- A usabilidade planejada para um produto pode ser definida, documentada e verificada (p.ex. como parte de um plano de qualidade).

Nesse caso, sendo então considerado o software como um produto e portanto, passível de uma avaliação de usabilidade usaremos conceitos da norma NBR 9241-11, para especificar e medir usabilidade.

3.3.1. Especificando e medindo a usabilidade de produtos(NBR ISO 9421-11)

Para se especificar ou medir a usabilidade se faz necessário a identificação dos objetivos, separar eficácia, eficiência e satisfação e os componentes do contexto de uso em subcomponentes com atributos mensuráveis e verificáveis.

A figura 2 ilustra a estrutura de usabilidade.

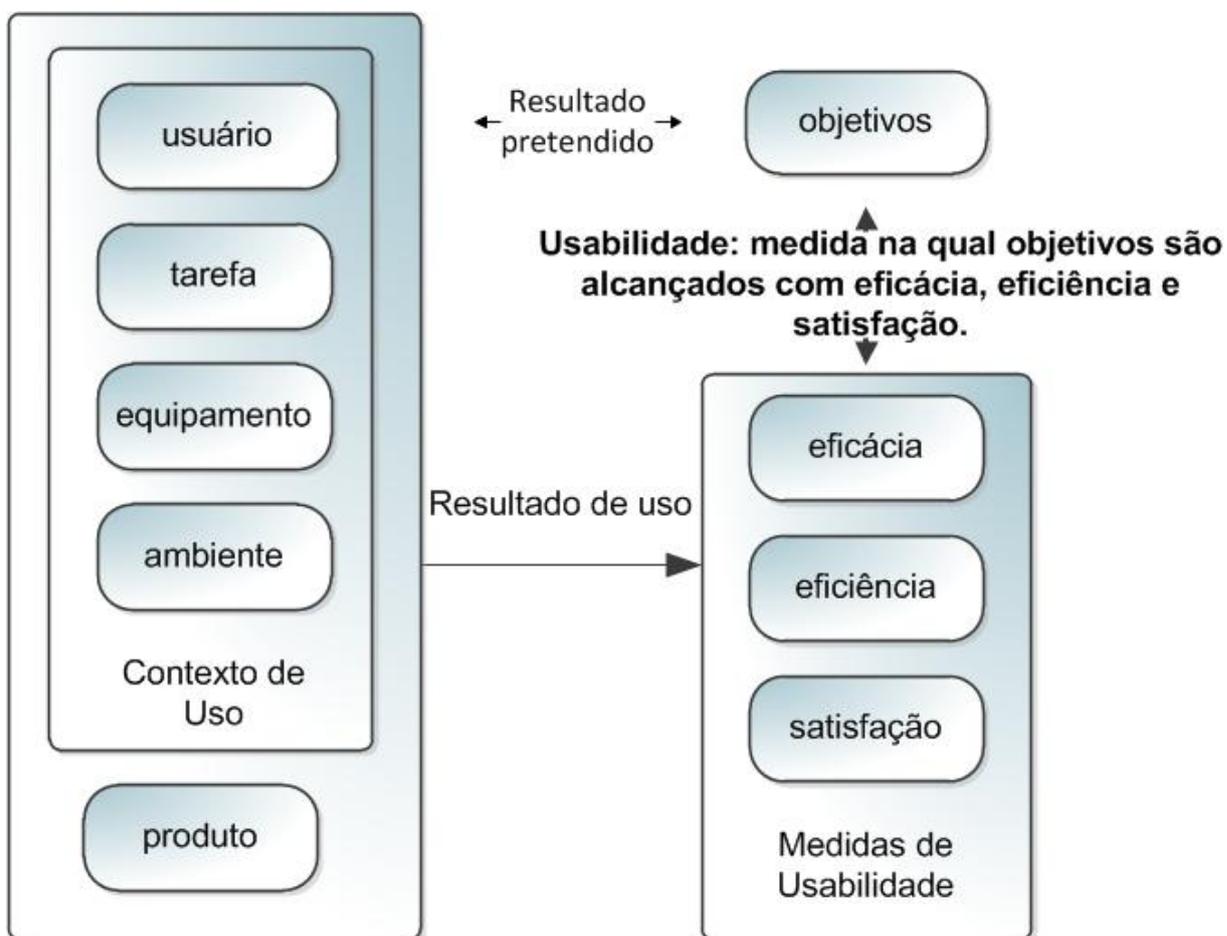


Figura 2 Estrutura de Usabilidade

Para especificar ou medir usabilidade são necessários:

- descrição dos objetivos pretendidos;
- um descrição dos componentes do contexto de uso (usuários, tarefas, equipamentos e ambientes);

- valores reais ou desejados de eficácia, eficiência e satisfação para os contextos pretendidos.

Normalmente é necessário fornecer pelo menos uma medida para eficácia, eficiência e satisfação, obviamente observando o contexto no qual a usabilidade está sendo descrita.

3.3.2. Desenvolvimento de produto

A definição e a estrutura para usabilidade podem ser usadas por equipes de desenvolvimento de produto para estabelecer um entendimento comum do conceito de usabilidade e podem ajudar a equipe de desenvolvimento de produto a determinarem a abrangência das questões associadas à usabilidade do produto.

A norma NBR ISO 9241-11 (2000) pode ser usada em conjunto com outras normas para especificar e medir os objetivos de usabilidade dentro de um projeto de desenvolvimento de software, o foco nesse projeto está nas interfaces e diálogos, os meios tradicionais de comunicação e feedback com o usuário final de um sistema.

3.3.3. Exemplo de medidas de usabilidade

A tabela traz um exemplo de medidas de usabilidade onde as medidas de usabilidade para eficácia, eficiência e satisfação são especificadas para objetivos globais.

Objetivos de Usabilidade	Medidas de eficácia	Medidas de eficiência	Medidas de satisfação
Usabilidade global	Porcentagem de objetivos alcançados;	Tempo para completar uma tarefa;	Escala de satisfação;
	Porcentagem de usuários completando a tarefa com sucesso;	Tarefas completadas por unidade de tempo;	Frequência de uso;
	Média da acurácia de tarefas completadas	Custo monetário de realização da tarefa	Frequência de reclamações

Tabela 8 Exemplo medidas de usabilidade

A tabela apresentada acima é um exemplo genérico que pode ser utilizado para testar a usabilidade de uma interface, através de alguns teste simples pode-se obter um feedback do usuário final e evitar futuros prejuízos, afinal de contas um software é utilizado justamente para otimizar tarefas, diminuir tempo e custo para as organizações que automatizam suas rotinas.

3.4. A INTERFACE GRÁFICA

Até certo tempo atrás, a importância da Interface Gráfica foi omitida durante o processo de desenvolvimento de software, o foco era simplesmente em um sistema funcional e também não existiam profissionais específicos para a área de “design”. Com o passar do tempo e a chegada de novas tecnologias para desenvolvimento, e com a necessidade de atingir usuários menos experientes e aumentar a interação com os sistemas computacionais um novo olhar foi lançado sobre a Interface Gráfica dos softwares.

Interface é uma superfície de contato que reflete as propriedades físicas das partes que interagem, as funções a serem executadas e o balanço entre poder e controle (ROCHA, 2000).

Segundo CYBIS (2007) a interface com o usuário é formada por apresentações, de informações, de dados, de controles e de comandos. É esta interface também que solicita e recebe as entradas de dados, de controles e de comandos. Finalmente, ela controla o diálogo entre as apresentações e as entradas. Uma interface tanto define as estratégias para a realização da tarefa, como conduz, orienta, recebe, alerta, ajuda e responde ao usuário durante as interações.

Uma boa interface torna o software atraente, intuitivo, eliminando ao máximo o medo de interação do usuário final. Incluídos nesse contexto entram os ganhos econômicos e produtivos resultantes de uma interface bem elaborada.

Nos dias atuais a Interface gráfica é um ponto crucial para as empresas de desenvolvimento que almejam um bom lugar no mercado, pois muitas empresas apresentam um sistema funcional e com uma Interface Gráfica bem atraente aos usuários finais.

Além da facilidade de uso, uma interface deve prover ao usuário um ambiente que não cause stress, cansaço ou repulsa. Para que isso ocorra um estudo sobre simetria, cores, figuras e agrupamento de informações faz-se parte básica do processo de desenvolvimento de uma interface.

“A interface gráfica é fator determinante no desenvolvimento de software porque para o usuário final a interface é o sistema”.

A figura a seguir adaptado de ROCHA (2000) ilustra o funcionamento da interface usuário-máquina.

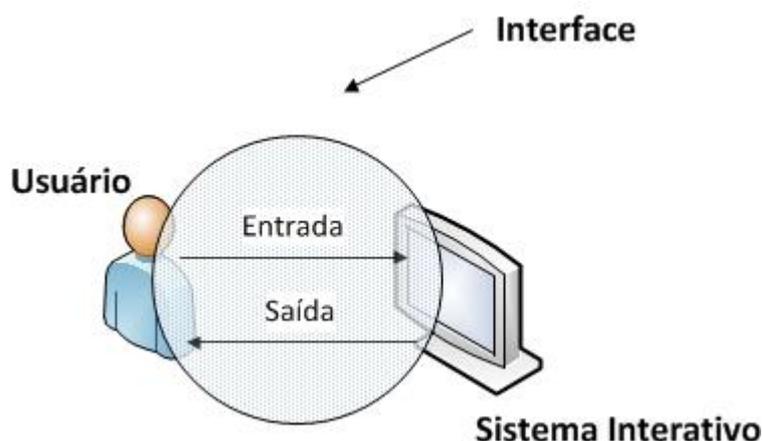


Figura 3 Funcionamento da Interface Usuário-Máquina (Rocha, 2000).

3.5. OTIMIZANDO A INTERFACE

Vários fatores são levados em consideração para se obter uma interface otimizada e na maioria das vezes são escolhas básicas que determinam a falta de qualidade de uma interface, como escolha de cores e distribuição de componentes na tela. Segundo Norman (1988) quatro princípios básicos devem ser observados:

- **Visibilidade e “affordances”:** apenas as coisas necessárias têm que estar visíveis, indicando quais as partes podem ser operadas e como o usuário interage com um dispositivo. Visibilidade indica o mapeamento entre ações pretendidas e ações reais, além de indicar também distinções importantes. A visibilidade do efeito da operação mostra se esta foi executada como pretendida. O que torna muitos dispositivos difíceis de serem operados é justamente a falta de visibilidade. Designers devem prover sinais que claramente indiquem ao usuário como proceder diante de uma determinada situação.

Affordances é o termo definido para se referir às propriedades percebidas e propriedades reais de um objeto, que deveriam determinar como ele pode ser usado. Quando em um sistema se tem a predominância da “affordance”, o usuário sabe que ação tomar somente olhando, sem a necessidade de figuras, rótulos ou instruções.

- **Bom modelo conceitual:** um bom modelo conceitual permite prever o efeito das ações. Sem este modelo bem definido o usuário efetua as operações solicitadas e não sabe que efeito esperar ou, o que fazer caso aconteça algo errado. O modelo conceitual é, portanto claro, e até óbvio, e exige um efetivo uso de “affordances”.

- **Bons mapeamentos:** relacionamento entre duas entidades é denominado de mapeamento. Especificamente em interfaces computacionais, indica o relacionamento entre os controles e seus movimentos e os resultados. Mapeamentos naturais levam ao entendimento imediato, desde que aproveitem analogia física e padrões culturais. Um objeto é fácil de ser usado quando existe um conjunto visível de ações possíveis, e os controles exploram mapeamentos naturais.

- **“Feedback”:** é o princípio básico de sistemas computacionais o retorno ao usuário sobre as ações que foram executadas e seus resultados obtidos. Sem um feedback o usuário sente-se “perdido”, sem a certeza de que sua solicitação foi executada, se ocorreu algum erro, se há a necessidade da execução de outro procedimento para continuidade do processo. É de grande importância a preocupação com o retorno da ação ao utilizador do sistema, transmitindo confiança, segurança e transparência do sistema.

3.5.1. Simetria

Correspondência, em grandeza, forma e posição relativa, de partes sitas em lados opostos de uma linha ou plano médio (Aurélio, 2004).

O equilíbrio simétrico se identifica pela disposição igual das imagens em torno dos eixos de equilíbrio localizados nos eixos centrais. O princípio da simetria descreve o agrupamento baseado nas propriedades emergentes da forma, ao invés das características das partes que a constituem. Em muitos casos, quanto maior a simetria de uma possível imagem ou de um gráfico, mais simples é visualizá-los e mais fáceis se tornam as interpretações sob vários aspectos (FERNANDES, 2009).

A simetria está presente no nosso cotidiano, onde o próprio ser humano é simétrico, a simetria também está presente na natureza, na engenharia, na

arte entre outros. Simetria sempre remete a ideia de equilíbrio, balanceamento, o que conseqüente traz uma sensação de segurança e conforto.

A figura 4 ilustra alguns exemplos de simetria na natureza.



Figura 4 Simetria na Natureza

A figura 5 ilustra exemplos da simetria na engenharia.



Figura 5 Simetria na Engenharia

Mais um exemplo de simetria, a figura 6 traz uma das obras mais famosas de Leonardo Da Vinci: o Homem Vitruviano.

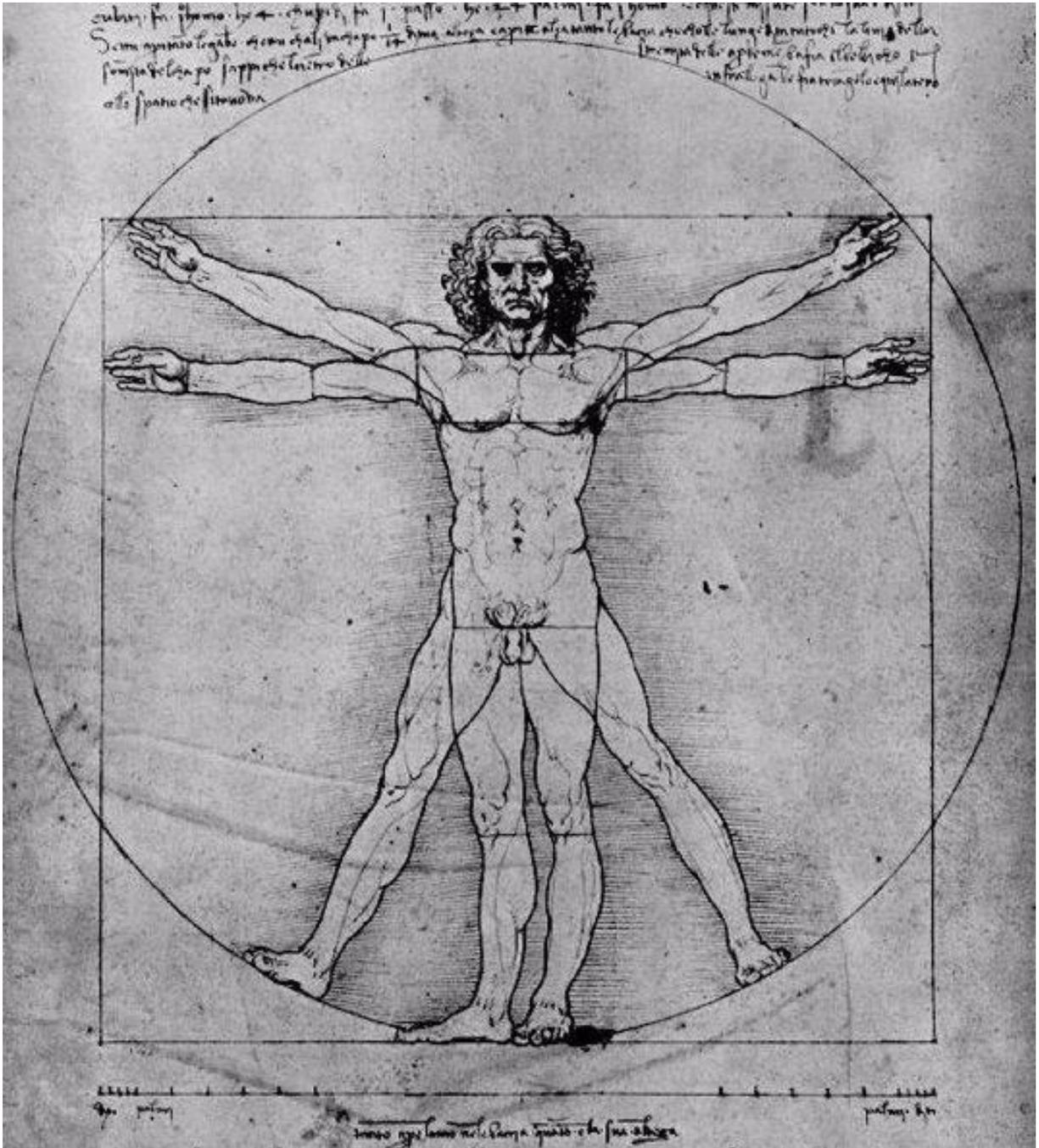


Figura 6 Simetria na Arte (Homem Vitruviano de Da Vinci)

Finalizando os exemplos de simetria, temos na figura 7 um “snapshot” da página inicial do “site” mais acessado do mundo, onde a simetria passa

praticamente despercebida, mais a hegemonia da página é percebida ao se traçar uma linha divisória, onde os botões de pesquisa se encontram simetricamente posicionados em relação à logomarca da empresa e o campo de pesquisa.

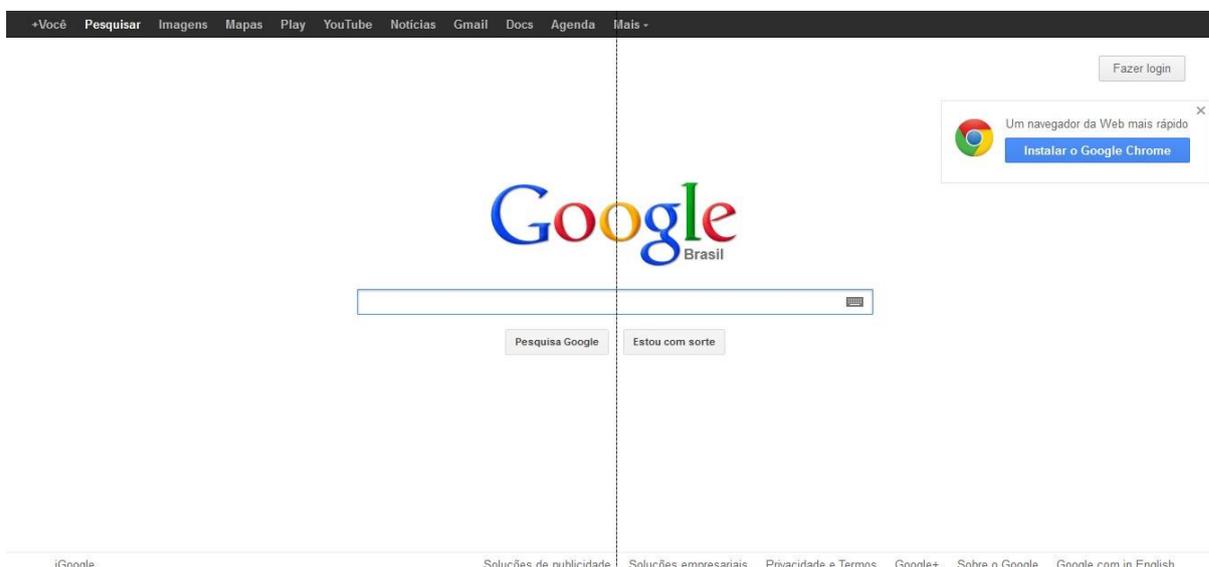


Figura 7 Snapshot Home Page Google

3.5.2. Cor

Sem sombra de dúvidas eis aqui um dos itens que compõem a interface gráfica que é o responsável pela inutilização de vários sistemas computacionais, a cor. Ela é peça fundamental na constituição de uma interface amigável (user-friendly) e também determina se uma interface passará nos testes de usabilidade.

A cor, elemento indissociável do nosso cotidiano, exerce especial importância, sobretudo nas Artes Visuais, na Pintura, Escultura, Arquitetura, Moda, Cerâmica, Artes Gráficas, Fotografia, Cinema, Espetáculo etc., onde é geradora de emoções e sensações (FONTANA, 2012).

“A reação dos indivíduos às cores se manifesta de forma particular e subjetiva, relacionada à vários fatores. Elas são estímulos psicológicos que influenciam no fato de gostar ou não de algo, negar ou afirmar, se abster ou agir. As

sensações sobre as cores se baseiam em associações ou experiências agradáveis ou desagradáveis” (ROVERI, 1996).

A Figura 8 apresenta um gráfico com o percentual dos sentidos do ser-humano no que diz respeito à absorção de informações do meio.

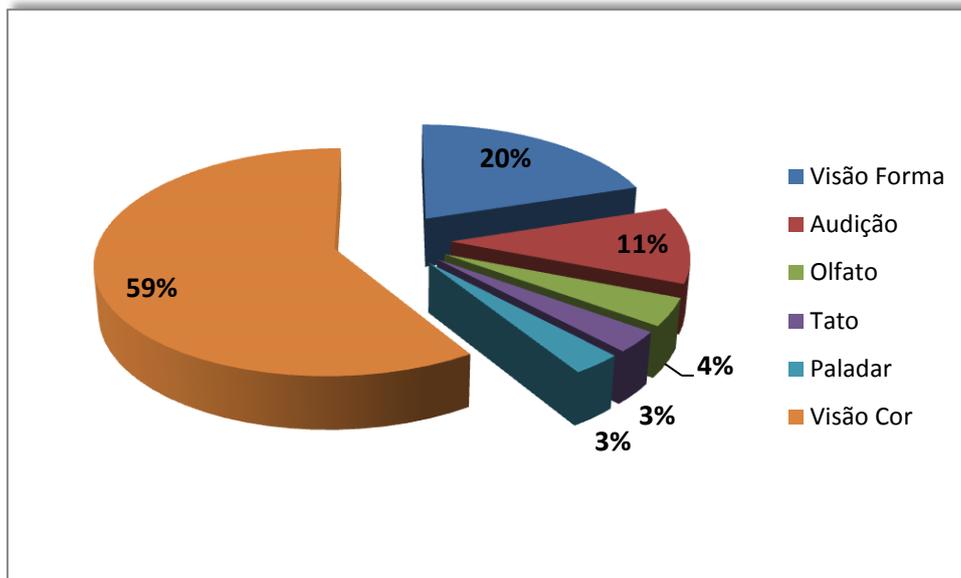


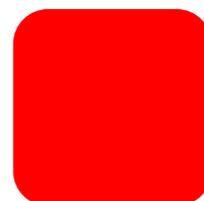
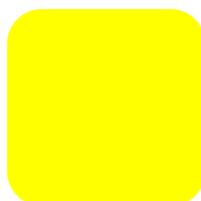
Figura 8 Percentual dos sentidos na captação das informações do meio

As cores são divididas em:

- Cores Primárias
- Cores Secundárias
- Cores Terciárias

3.5.2.1. Cores Primárias

As cores primárias e também conhecidas como as "cores puras", são originárias de pigmentos naturais (vegetal e mineral). São elas: Azul, Amarelo e Vermelho.



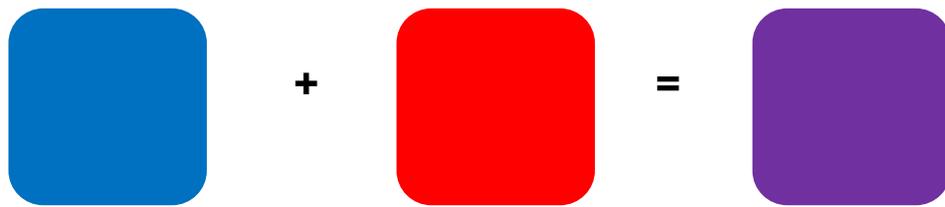
A teoria das cores diz que por meio de cores básicas, ou primárias, qualquer cor pode ser formada.

3.5.2.2. Cores Secundárias

As cores secundárias são o resultado da mistura de duas cores primárias, na mesma proporção.

Fórmula = (primária + primária) = secundária

Ex:

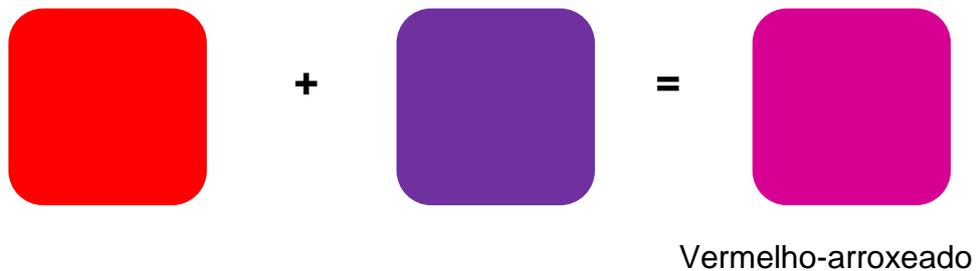


3.5.2.3. Cores Terciárias

As Cores Terciárias são o resultado da mistura de uma cor primária com uma cor secundária.

Fórmula = (primária + secundária) = terciária

Ex:



Também podemos dividir as cores pela sua “temperatura”:

- Cores quentes
- Cores Frias

3.5.2.4. Cores Frias

São aquelas cores onde existe a predominância dos tons de azul, verde e roxo. São caracterizadas como cores tristes e melancólicas, dando a sensação de calma e tranquilidade.

3.5.2.5. Cores Quentes

São aquelas cores onde existem a predominância dos tons de vermelho, amarelo e laranja. São caracterizadas como cores alegres e vibrantes e que nos dão a sensação de calor.

3.5.2.6. Cores Neutras

São aquelas cores onde não existem a predominância nem dos tons quentes e nem dos tons frios. Estas cores servem como base para qualquer cor, sejam elas de qualquer tonalidade. Preto, Cinza, Branco, Marrom e Bege (FONTANA, 2012).

3.5.2.7. Significado das cores

- **Preto:** está associado à ideia de morte, luto ou terror, no entanto também se liga ao mistério e à fantasia, sendo hoje em dia a cor da sofisticação e luxo.
- **Branco:** associa-se à ideia de paz, de calma, de pureza. Significa inocência e pureza.

- **Cinzeno (Cinza):** pode simbolizar o medo ou a depressão, mas é também uma cor que transmite estabilidade, sucesso e qualidade.
- **Bege:** é uma cor que transmite calma e passividade. Está associada à melancolia e ao clássico.
- **Vermelho:** é a cor da paixão e do sentimento. Simboliza o amor, o desejo, mas também simboliza o orgulho, a violência e agressividade.
- **Verde:** significa vigor, juventude, frescor, esperança e calma.
- **Amarelo:** transmite calor, luz e descontração. É também uma cor energética, ativa que transmite otimismo. Está associada ao Verão.
- **Laranja:** é uma cor quente, tal como o amarelo e o vermelho. Significa movimento e espontaneidade.
- **Azul:** é a cor do céu, do espírito e do pensamento. Simboliza a lealdade, a fidelidade, a personalidade e subtileza. Simboliza também o ideal e o sonho.
- **Castanho (Marrom):** é a cor da Terra. Esta cor significa maturidade, consciência e responsabilidade. Está ainda associada ao conforto, estabilidade, resistência e simplicidade.
- **Roxo:** transmite a sensação de tristeza. Significa prosperidade, nobreza e respeito.
- **Lilás:** significa espiritualidade e intuição.
- **Rosa:** significa beleza, saúde, sensualidade e também romantismo.
- **Prata:** é uma cor associada ao moderno, às novas tecnologias, à novidade, à inovação.
- **Dourado:** está simbolicamente associado ao ouro e à riqueza, a algo majestoso. (FONTANA, 2012)

3.5.2.8. O uso das cores na Interface

A cor é o elemento visual da interface que mais influencia na usabilidade de um software, pois ela interfere diretamente na aparência, legibilidade, fadiga (no caso de uso contínuo) entre outros.

Recomenda-se cuidado com o uso indiscriminado da cor. É aconselhável que primeiro se faça o projeto em Preto e Branco e depois o colora com cuidado, usando cores neutras. Use poucas cores e com mesma luminância (brilho) e utilize cores brilhantes com cautela. Existem teorias que defendem que as cores causam sensações às pessoas (OLIVEIRA, 2007).

Sommerville (2003) recomenda não mais que sete cores para um sistema.

O uso de cores em interfaces permite:

- mostrar as coisas conforme são vistas na natureza
- representar associações simbólicas
- chamar e direcionar a atenção
- enfatizar alguns aspectos da interface
- determinar um estado de espírito
- auxiliar na identificação de estruturas e processos
- diminuir a ocorrência de erros
- tornar uma interface mais fácil de ser memorizada (MAIA, 2012)

Porém dentro de um projeto de interface os seguintes cuidados também devem ser tomados:

- uma escolha não adequada de cores pode interferir na legibilidade da interface
- as cores podem apresentar características distintas em condições diferentes
- as cores devem ser selecionadas de modo a não causarem fadiga nos olhos do usuário e nem deixá-lo confuso
- o grupo de pessoas a quem se destina o sistema deve ser considerado; algumas comunidades podem apresentar reações arbitrárias porém negativas diante de certas cores

- alguns elementos da tela são agrupados com as mesmas cores; deve-se tomar um cuidado para não se agrupar elementos que não se relacionam entre si (MAIA, 2012).

As cores acromáticas (os vários tons de cinza, situados entre o branco e o preto) são as mais indicadas para o fundo das interfaces. O branco nem sempre é o mais indicado devido ao seu brilho intenso, o preto não é indicado pois traz a sensação que os objetos estão “soltos” ou pendurados, o cinza seria o ideal, mais deve ser usado com moderação pois traz a aparência de interfaces antigas, com a disponibilidade do controle de luminosidade nos equipamentos atuais (monitores, celulares, etc.) o uso do branco como background das interfaces é predominante, e se faz notar presente na maioria das aplicações atuais.

A figura 9 traz um snapshot da home page do canal de notícias CNN, trazendo um “background” na cor branca.



Figura 9 Snapshot Home Page CNN (Background na cor branca)

A figura 9 traz um “print” de um formulário de cadastro da mega rede de vendas “online” Walmart, também trazendo o “background” na cor branca.

Identificação - Novos Usuários

[Problemas com seu cadastro?](#) [Central de Atendimento](#) | [Ajuda ao vivo](#)

Dados de acesso ao Walmart

Pessoa Física Pessoa Jurídica

Seus dados pessoais:

Nome*:

Sobrenome*:

Como gostaria de ser chamado:

CPF*: [Por que Pedimos seu CPF](#)

Data de Nascimento*: dd/mm/aaaa

Sexo*: Masculino Feminino

CEP*: [Não sei meu CEP](#)

Telefone de Contato 01*:

Figura 10 Print Formulário Cadastro de Usuário (Rede Walmart)

O Facebook, o maior site de relacionamentos da atualidade também utiliza o branco como plano de fundo. Não é recomendado que se utilizem cores ou figuras como background pois podem confundir ou tirar a atenção do usuário.

As cores quentes são mais efetivas quando se deseja chamar a atenção, o vermelho, por exemplo, está associado à situação de perigo.

A figura 11 apresenta um ícone de exclusão, predominante na cor vermelha, o usuário já saberá que se trata de uma situação de risco, no caso a exclusão de dados, o que pode ser um processo irreversível.



Figura 11 Ícone de exclusão (Cor vermelha)

O verde e o azul geralmente estão associados a situações que não remetem perigo ou atenção, por exemplo, a opção de salvar ou atualizar dados em um sistema não é um processo irreversível, portanto é associado às cores verde ou azul.

A figura 12 ilustra um ícone para o opção de salvar, com a cor verde predominante. É importante que o desenvolvedor se atente a algumas

convenções que já são padrões no mercado, pois não foram adotadas por acaso e já fazem parte do cotidiano dos usuários de sistemas computacionais.



Figura 12 Ícone Salvar (Cor verde)

A figura 13 traz uma curiosidade sobre a logomarca de duas grandes empresas da área de tecnologia, ambas apresentam as mesmas cores na sua composição.



Figura 13 Logomarca Google e Microsoft (Mesmas cores na composição)

O uso da cor é de fundamental importância para o desenvolvimento de Interfaces Gráficas, o uso adequado da cor torna o ambiente atraente e agradável para o usuário final.

3.5.3. Ícones

São componentes da Interface gráfica que falam por si só, trabalham em um conjunto que agrega cores e símbolos. Essa combinação faz com que o usuário saiba o que esperar ao clicar, tocar ou selecionar um ícone. Nos dias atuais, com o crescimento da tecnologia “touchscreen” (tela sensível ao toque), aliado ao aumento de uso de telefones celulares e “tablets” os ícones ganharam um grande destaque.

Os ícones devem ser significativos, apropriados, coerentes, consistentes, claros, simples e definidos em pequeno número. Seu tamanho deve ser econômico em relação ao espaço de tela. Dependendo de sua utilização aconselha-se a adoção de bordas bem definidas. É recomendável utilizar sempre ícones prontos, respeitando seus significados (OLIVEIRA, 2007).

A figura 14 traz a imagem de um usuário interagindo com uma tela touchscreen, onde somente ícones são usados como referência para acesso às funções do aparelho.



Figura 14 Interação Usuário (“touchscreen” interação com ícones)

Alguns pontos devem ser relevados durante o processo de construção de um ícone:

- A imagem associada aos ícones deve ser representativa da função que lhe é atribuída. E mais, esta deve ser familiar à linguagem do público-alvo.
- Ícones em formato de botões devem ter dimensões compatíveis com as dos dedos dos usuários.
- Ícones devem conter simplicidade, cores suaves e design visual identificável, para que possam evitar conflitos de interpretação e ambigüidades.
- Ícones devem vir sempre que possível acompanhados de show hint (descrição textual associada a eles).

A figura 15 traz alguns exemplos de ícones tradicionais que já fazem parte do dia a dia dos usuários de sistema computacionais; O primeiro ícone da esquerda para direita (caracter de interrogação) é associação à ajuda (help); O segundo ícone (meio) com a figura de uma lupa remete a idéia de busca (search) e o terceiro e último ícone (esquerda para a direita) com um símbolo de uma notação musical faz referência à um aplicativo de música em um celular. O objetivo do ícone é justamente ser o mais intuitivo possível.



Figura 15 Exemplos de ícones (Ajuda, Busca e Música).

3.5.4. Fontes

O uso correto da fontes também é um grande diferencial na construção de uma boa interface gráfica. Assim como nos outros itens a cor utilizada nos elementos textuais da Interface também interferece na usabilidade da mesma.

Quanto ao tamanho da fonte, não utilizar menor que 10 pontos, limitando o uso de fontes diferentes para textos (até dois tipos). Deve-se evitar também fontes muito grandes, que pareçam “gritar” com o usuário. Evitar textos só com maiúsculas, e não exagerar com o sublinhado, negrito e itálico (OLIVEIRA, 2007).

O ideal é que se mantenha uma padronização das fontes durante o desenvolvimento da Interface.

3.5.5. “Layout”

O “layout” ou a forma como os componentes são dispostos na interface (janela, caixa de diálogo, mensagem), também facilitam a compreensão das funcionalidades do sistema por parte do usuário.

Algumas características básicas podem fazer uma grande diferença na construção de um bom layout para uma interface gráfica. A seguir alguns pontos que devem ser observados nessa etapa da construção da interface:

- é ideal que se mantenha um padrão entre todas as telas do sistema (padronização);
- evitar espaços vazios ou carregados de componentes ou informações;
- manter os componentes da interface alinhados (campos de texto, rótulos, botões);
- agrupar informações por tipo e importância.

A figura 16 traz o formulário de cadastro de usuários de uma gigante de vendas “online” (Americanas.com), nesse exemplo um bom “layout” é definido com alinhamento de todos os componentes e agrupamento de informações. No caso Cadastro de Pessoa Física e Cadastro de Endereço ficam separados. Esta prática de agrupamento de informações facilita em muito o preenchimento de formulários. É pertinente que, após digitar a rua onde mora, seja solicitado número, bairro, ou seja informações que ligam-se umas as outras, não fazendo sentido que estejam desconexas.

Nessa mesma figura pode-se observar mais bons exemplos de uma boa interface: a utilização do “background” em branco (cor neutra); as fontes no tom cinza que também é uma cor neutra; no cabeçalho do site pode-se notar alguns ícones que simbolizam os vários departamentos da loja; a cor vermelha nesse caso está intimamente ligada à marca da “megastore” (Americanas.com), portanto não influenciará na navegação do usuário pelo site.

americanas.com buscar minha conta 0 itens

veja todos os 36 departamentos

Identificação

*Campos obrigatórios.

CADASTRO PESSOA FÍSICA

Pessoa Física Pessoa Jurídica

Nome completo*

CPF*

Sexo*

Data de Nascimento*

Telefone Preferencial* -

Telefone Alternativo -

Telefone Comercial -
(DDD - TELEFONE) (OPCIONAL)

Como gostaria de ser chamado?*

E-mail*

Senha*

Confirma senha*

Deseja receber e-mail de ofertas e promoções da Americanas.com? sim

CADASTRO ENDEREÇO

CEP*

Identificação do Endereço* Apartamento Casa Comercial Outro

Figura 16 Formulário cadastro de usuários (Americanas.com)

3.6. ESTUDOS DE CASO (INTERFACES FALHAS EXISTENTES NO MERCADO)

Esse capítulo trará “snapshots” de alguns sistemas que encontram-se à venda na internet e também de alguns sites que encontram-se em funcionamento até a presente data. Vale ressaltar que esses snapshots serão utilizados como objetos de estudo.

Na figura 17 pode-se listar as seguinte falhas na construção da interface:

“Background”: a cor do “background” apesar de se enquadrar entre as cores neutras, não é agradável, e remete a uma idéia de um sistema já ultrapassado; também cria confusão com a cores dos campos de texto.

Excesso de informações: existe um excesso de informações no formulário; como os itens não foram agrupados por tipo e/ou informação o preenchimento das informações tende a ficar mais confuso.

Ícones: os ícones não são claros e objetivos (não conseguem transmitir uma idéia concreta do que representam).

Fonte: a fonte escolhida e seu espaçamento torna difícil a leitura dos textos.

Código	Nome / Razão Social	Data	
001	EngWhere Orçamentos Ltda	01/09/2010	
1º Endereço		Bairro	
Rua José Urbano, 180		Centro	
Cidade	U.F.	Cep	CNPJ / CPF
São Tomás de Aquino	MG	37960-000	
Telefone	Fax	Contato(s)	Classificação
(035) 3535-1759	(035) 3535-1759	Amado / Dora / André	Parceiro
E-mails		URL	
engwhere@engwhere.com.br		www.engwhere.com.br	
2º Endereço		U.F.	Cep
Praça Cônego Tomás, nº 2 - Centro		MG	37960-000
Observações			
Softwares: EngWhere Orçamentos / Previso / BDI / Geedoc / Gercon			

					
Novo	Apaga	Atualiza	Imprimir	Ajuda	Fechar

	Código	Nome / Razão Social	Data	Endereço I	Bairro / Complemento
▶	001	EngWhere Orçamentos Ltda	01/08/2008	Rua José Urbano, 180	Centro

Figura 17 Snapshot Clientes e Fornecedores EngWhere

(http://www.engwhere.com.br/webhelp/index.htm#page=ajuda_htm/Cadastro_clientes_fornecedores.htm)

Na figura 18 o desenvolvedor peca somente no “background”, os ícones possuem boa aparência e delimitação, a fonte também é legível . A cor utilizada no “background”, além de se tornar cansativa com o passar do tempo, também dificulta a visualização dos ícones.



Figura 18 Menu Campus Sistema de Integração Factoring
 (http://campusinformatica.com.br/Help/Factoring/Hlp_foCadastroClientes.html)

A figura 19 traz um cadastro de clientes que a primeira vista choca pela quantidade de informações exibidas. Entre outros itens pode-se observar:

“Background”: a cor do background se enquadra entre as cores neutras, mais não se equilibra com as outras cores existentes na interface.

Agrupamento das Informações: as informações estão agrupadas, mas o espaçamento e a maneira como elas foram agrupadas não permitem uma fácil distinção.

Ícones: alguns ícones não remetem a sua verdadeira função (símbolo tradicional de recortar/cut é usado para opção de excluir); em outros ícones fica praticamente impossível interpretar o significado.

Fonte: o espaçamento entre os caracteres torna difícil a leitura; a cor das fontes nos campos de texto é inapropriada, além do preenchimento em caixa alta que traz uma sensação desagradável ao usuário.

Figura 19 Cadastro de Cliente Fidelit-Pharmanet
(<http://www.pharmanet.com.br/fidelity.htm>)

A figura 20 é um exemplo clássico de um software que traz uma interface que provavelmente seria reprovada em todos os testes de usabilidade.

“Background”: a cor do “background” é completamente desagradável, causando repulsa à primeira vista.

Excesso de informações: a quantidade de informações no formulário e sua distribuição tornam difícil tanto o preenchimento quanto o entendimento dos dados na sua recuperação.

Ícones: os ícones não são claros e não podem servir de base para uma tomada de decisão dentro do sistema.

Fonte: a fonte escolhida e seu espaçamento torna difícil a leitura dos textos, o preenchimento em caixa alta transmite uma sensação desagradável e ajuda a aumentar o excesso de informações contidas no formulário.

Informações sobre o Cliente MAXI AÇÃO MARKETING INFORMATICA

[DADOS DO CLIENTE] Pessoa Jurídica 2

R F V Razão Social: MAXIAÇÃO MARKETING INFORMATICA Codinome/Fantasia: MAXIAÇÃO

CNPJ: 12.345.678/0001-20 Inscrição Estadual: 123 Matriz Filial Loja: 001

CEP: 09060-500 Logradouro: R Fundação: 15/10/1991

Endereço: GRA-BRETANHA Número: 304

Complem: CASA Situação Financeira:  V.I.P.

Bairro: VILA PRINCIPE DE GAL Estado: SP

Cidade: SANTO ANDRE Carga: DIRETOR

Pais: Bra Contato: JOSE CARLOS Remal: FAX: 45552816

DDD: 11 DDI: Tel./ Cel: 45552599 49943131

Representante: ADMMAXI

Tipo: Provel Origem: TMKT

Consulta: E-Mail: comercial@maxiacao.com.br Web: www.maxiacao.com.br

Consultar	Contato	Histórico	Ok Data	Hora	Agendado para	Observação
ADMMAXI	Supervisão	AT Atualização Cadastral	09/05/2006	16:27:39	24/05/2006	AT Atualização Ca

Empal Fatura Cobrança Perfil Editor Recibe Novo Consulta Altera Inclui Ligação Exclui Volta

Figura 20 Cadastro de Clientes Maxi Ação
(<http://www.maxiacao.com.br/si/site/0613>)

Na figura 21 temos um exemplo de um formulário de fechamento de venda na internet, o excesso de informações e de cores, torna difícil a compreensão e a tomada de decisão. A falta de alinhamento dos campos também traz uma sensação de desorganização.

O plano de fundo do “site” na cor rosa e com figuras repetidas também produz um efeito negativo. Vale lembrar que no caso de um site de vendas a aparência é de fundamental importância para que um usuário venha a finalizar uma compra.

The screenshot shows the checkout process on the Centertel website. At the top, there are promotional banners for 'BOLETO POS-PAGO' and 'BRINDES NA 1ª COMPRA DE QUALQUER VALOR'. Below these is a search bar and navigation links. The main content area features a table with the following data:

Foto	Descrição	Quantidade	Unit.	Total	Opção
	Telefone Celular de Noss	1	298,00	298,00	EXCLUIR

Below the table, there are buttons for 'Continuar compra' and 'Cancelar compra', and a 'PARCIAL: R\$ 298,00' field. Further down, there is a form for 'Seu e-mail' and 'Para cálculo do Frete, digite seu CEP aqui: 19807535'. A shipping selection section asks to choose a form of delivery for 'ASSIS - SP', with options for 'Sedex' (R\$ 32,86), 'Envio Normal' (R\$ 16,96), and a 'Braspress' option (R\$ 0,00) which is noted as unavailable. The 'VALOR FINAL DO PEDIDO' field is empty. At the bottom, there are more 'Continuar compra' and 'Cancelar compra' buttons, and a visitor ID: 'Você é o visitante nº 1206539'.

Figura 21 Fechamento de venda Centertel (www.centertel.com.br)

4 Desenvolvimento

Neste capítulo será feita uma breve abordagem sobre a tecnologia escolhida para o desenvolvimento do material de apoio, bem como os números que comprovam o crescimento dessa tecnologia no mercado atual.

Também será descrito o processo da criação do material de apoio resultante dos estudos desse projeto, ressaltando que o foco desse trabalho é o desenvolvimento de boas interfaces gráficas (interação com o usuário e apoio ao desenvolvedor), portanto algumas etapas do desenvolvimento como etapas avançadas de codificação serão omitidas, mais estarão disponíveis através da documentação gerada (JavaDoc) e do código-fonte assim como todo material que estará disponível como anexo desse projeto.

Para esse projeto a tecnologia de desenvolvimento escolhida foi a linguagem de programação Java, em conjunto com a IDE de Desenvolvimento Netbeans, mas outras linguagens existentes no mercado fornecem todas as condições para o desenvolvimento de boas interfaces.

4.1 JAVA

Java é uma linguagem de programação e uma plataforma de computação lançada pela primeira vez pela Sun Microsystems em 1995. É a tecnologia que capacita muitos programas da mais alta qualidade, como utilitários, jogos e aplicativos corporativos, entre muitos outros, por exemplo. O Java é executado em mais de 850 milhões de computadores pessoais e em bilhões de dispositivos em todo o mundo, inclusive telefones celulares e dispositivos de televisão. (Java.com, 2012).

A seguir algumas características da linguagem de programação Java:

- **Write once, run everywhere:** escreva uma vez e execute em qualquer lugar. Este é o motor de Java, o que impulsionou a linguagem, a característica de compatibilidade do código Java com qualquer hardware. A promessa que navegadores web, telefones, PDAs e outros aparelhos eletrônicos pudessem executar um mesmo código fonte Java ajudou a popularizar a linguagem e atrair novos programadores;
- **Web programming:** a Sun fez um prognóstico arrojado para a época afirmando que a web era um computador logo, deveriam existir ferramentas para “programar” este novo dispositivo, a rede. A arquitetura cliente/servidor é algo quase que natural em Java;
- **Recursos de Rede:** possui extensa biblioteca de rotinas que facilitam a cooperação com protocolos TCP/IP, como HTTP e FTP;
- **Segurança:** permite que usuários utilizem softwares seguros que foram copiados da web;
- **Programação dinâmica e extensível:** a programação Java é baseada integralmente em POO (Programação Orientada á Objetos), ou seja, é baseada na descrição de classes. Para conceitos novos e tarefas novas criam-se novas classes ou, realizam-se a manutenção ou incrementam-se classes antigas;
- **Eficiência na codificação:** com a utilização de classes, a programação torna-se mais eficiente já que as chances de reescrever software inédito tornam-se menor, ou seja, sempre existirá uma classe que faz aquilo que o programador deveria implementar;
- **Internacionalização:** é o poder de escrever código fonte que trabalhe com caracteres dos mais diferentes alfabetos. Este foi um dos últimos recursos, dos citados aqui, a ser implementado em Java;

- **Garbage collector:** objetos não utilizados são retirados da memória principal automaticamente. Isto se chama deslocamento automático de memória dinâmica;
- **Concorrência:** permite o uso de threads múltiplos e mecanismos explícitos de concorrência;
- **Tratamento de eventos e exceções:** permite a programação orientada por eventos e a possibilidade de escrever programas que respondem a falhas de execução;
- **Carga Dinâmica de Código:** programas em Java são formados por uma coleção de classes armazenadas independentemente e que podem ser carregadas no momento de utilização.

Até hoje, a plataforma Java já atraiu mais de 9 milhões de desenvolvedores de software. É utilizada em todos os principais segmentos da indústria, estando presente em uma ampla gama de dispositivos, computadores e redes.

A tecnologia Java está presente em mais de 4,5 bilhões de dispositivos, incluindo:

- Mais de 800 milhões de PCs
- 2,1 bilhões de telefones celulares e outros dispositivos portáteis
- 3,5 bilhões de cartões inteligentes
- Além de set-top boxes, impressoras, webcams, jogos, sistemas de navegação para automóveis, terminais lotéricos, dispositivos médicos, guichês de pagamento de estacionamento etc. (Java.com, 2012).

4.2 DESCRIÇÃO (INTERFACEGRAFICADPA)

Nesse trabalho foi desenvolvido um material de apoio com base nas metodologias, normas e conceitos adquiridos no decorrer do projeto. O objetivo de facilitar a vida do usuário final e apoiar o desenvolvedor resultou em um material denominado de InterfaceGraficaDPA composto pelos seguintes elementos:

- Biblioteca de componentes (Swing Desktop) adaptáveis às paletas das IDE's Netbeans e Eclipse.
- Aplicativo (InterfaceGraficaDPAPreview) que gerencia a exibição dos componentes, exemplos e questionário.
- Questionário para uma análise rápida de itens essenciais para uma boa interface Gráfica (com base nos conhecimentos obtidos no projeto).
- Protótipos de Exemplo.
- Formulários (Exemplos e Dicas).

4.3 INTERFACEGRAFICADPA (BIBLIOTECA SWING)

A biblioteca InterfaceGraficaDPA foi desenvolvida utilizando a linguagem de programação Java em conjunto com a IDE Netbeans. As cores, formas e tamanhos dos componentes foram baseados na Fundamentação Teórica e Estudos de caso desse respectivo trabalho .

InterfaceGraficaDPA é um biblioteca de componentes Swing para desenvolvimento Desktop (personalizados) adaptáveis à paleta das IDE's de desenvolvimento Netbeans e Eclipse. Tem o intuito de ajudar os desenvolvedores com componentes pré-configurados com cores e fontes adequadas. O uso da biblioteca InterfaceGraficaDPA em um projeto pode proporcionar maior agilidade e padronização dos componentes da Interface.

Após compilado o projeto é gerado um arquivo com a extensão .jar (InterfaceGraficaDPA.jar), as IDE's de desenvolvimento provem a facilidade de adicionar esse arquivo jar a paleta de componentes o que possibilita ao usuário (desenvolvedor) clicar e arrastar os componentes para a implementação das Interfaces.

Na figura 22 temos um print que exhibe a InterfaceGraficaDPA adicionada a paleta da IDE Netbeans.

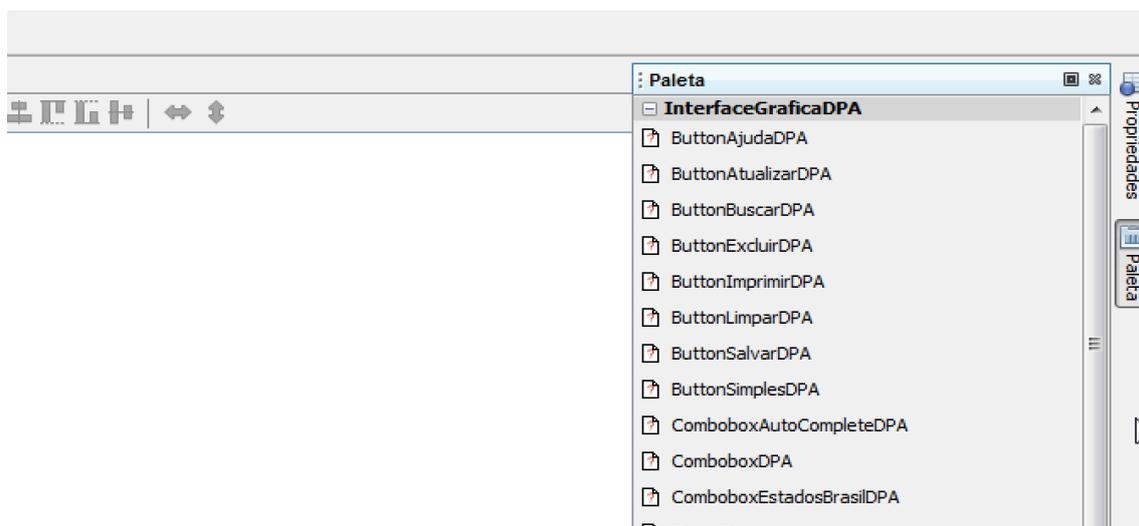


Figura 22 - Print InterfaceGraficaDPA na paleta do Netbeans

Na figura 23 podemos observar que ao se utilizar um componente da Biblioteca InterfaceGraficaDPA automaticamente o pacote Jar (InterfaceGraficaDPA.jar) é adicionado as bibliotecas do projeto.

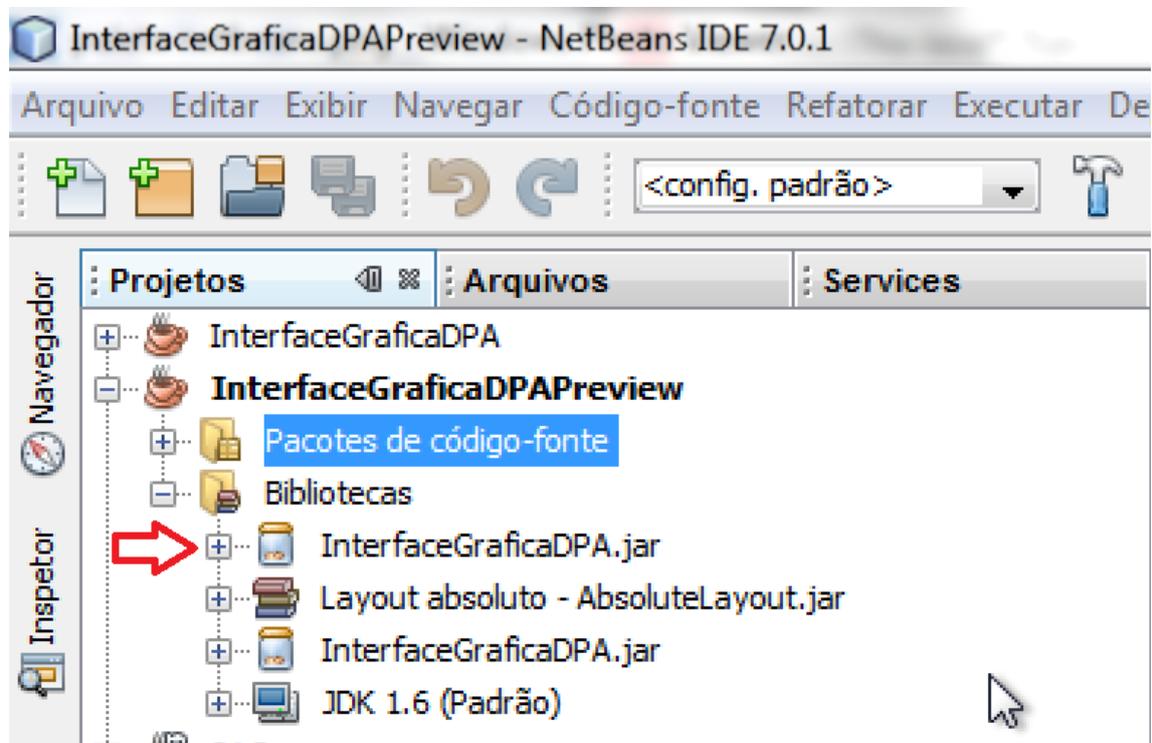


Figura 23 - Biblioteca InterfaceGraficaDPA adicionada as bibliotecas de um projeto (Netbeans)

A figura 24 mostra biblioteca InterfaceGraficaDPA adicionada a paleta da IDE Eclipse.

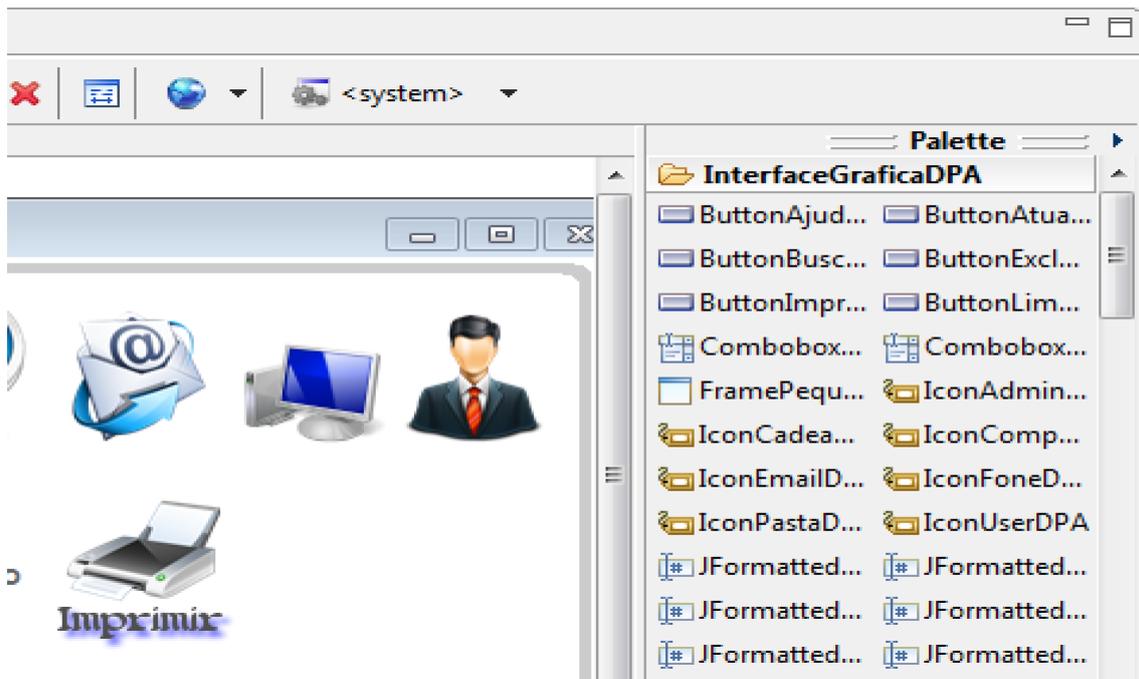


Figura 24 - InterfaceGraficaDPA adicionada a paleta da IDE Eclipse

Assim como na IDE Netbeans assim que um componente da biblioteca InterfaceGraficaDPA é adicionado a uma Interface o arquivo InterfaceGraficaDPA é adicionado as bibliotecas do projeto com mostra a figura 25.

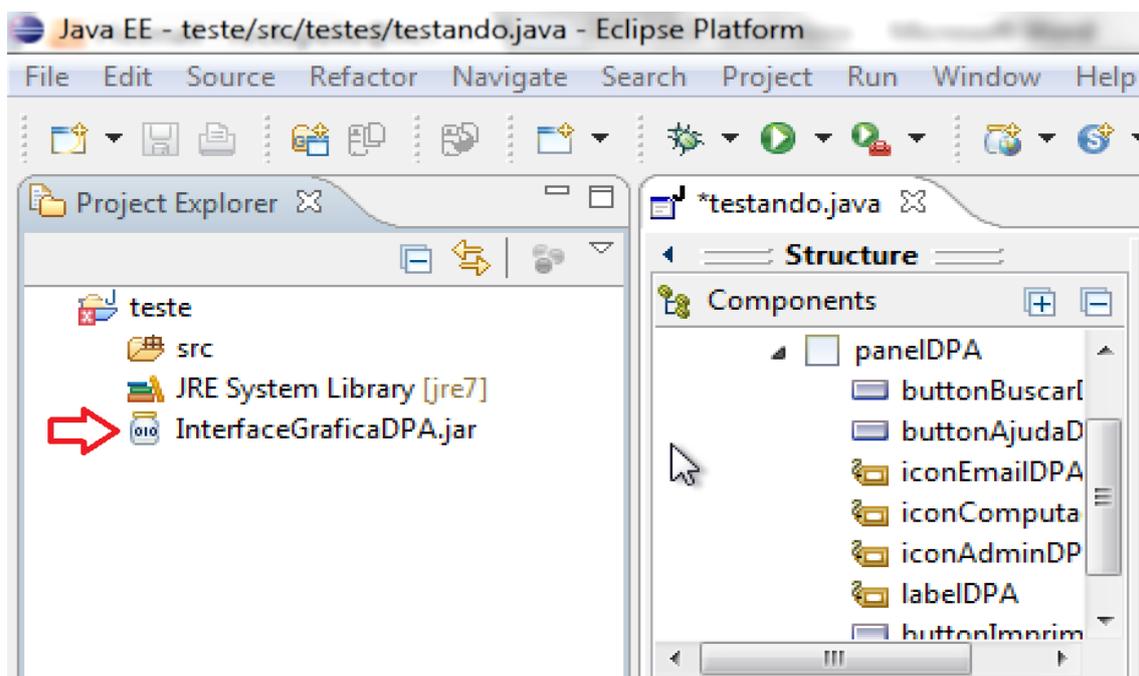


Figura 25 - Biblioteca InterfaceGraficaDPA adicionada as bibliotecas de um projeto (Eclipse)

4.3.1 InterfaceGraficaDPA (Conceitos Java)

Para a implementação e conseqüentemente a personalização dos componentes da Biblioteca foram utilizados conceitos de Herança (Extends) e Interface (Implements).

Abaixo são especificados os conceitos básicos que permitem personalizar componentes (que também são classes), através da linguagem Java:

- **Herança** : Uma classe que é derivada de uma outra classe é chamada uma subclasse (também uma classe derivada , classe estendida , ou classe filha). A classe da qual é derivada da subclasse é chamada de superclasse (também uma classe base ou classe pai). Exceto *Object* , que não tem superclasse, cada classe tem uma e apenas uma (herança simples) superclasse direta. Na ausência de qualquer outra superclasse explícita, cada classe é implicitamente uma subclasse de *Object*. Classes podem ser derivados a partir de classes que são derivados a partir de classes que são derivados a partir de classes, e assim por diante, e em última análise, derivado a partir da extremidade superior da classe, *Object* . Essa classe é dito ser descendente de todas as classes na cadeia de herança que remetem a *Object*.

A classe *Object*, definida no `java.lang` pacote, define e implementa o comportamento comum a todas as classes incluindo as que você escreve. Na plataforma Java, muitas classe decorrem diretamente de *Object*, outras classes derivam de algumas dessas classes, e assim por diante, formando uma hierarquia de classes

A figura 26 traz a hierarquia de classes Java onde todas as classes descendem de *Object*.

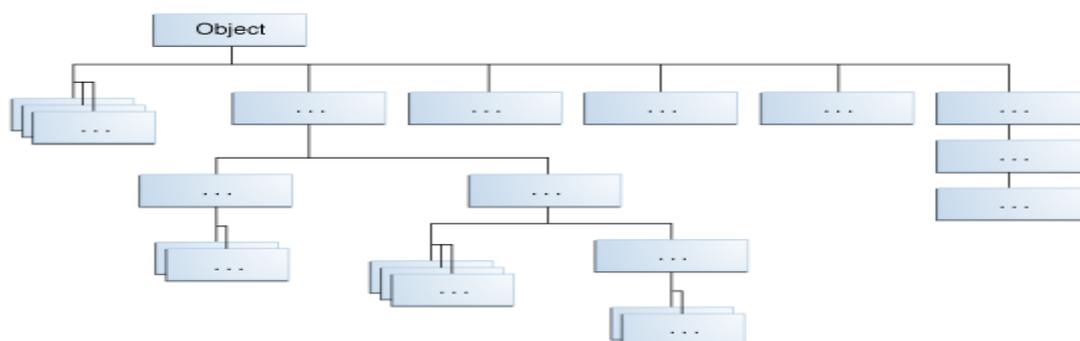


Figura 26 - Hierarquia de Classes Java

A ideia de herança é simples, mas poderosa: Quando você quiser criar uma nova classe e já existe uma classe que inclui parte do código que você quiser, você pode derivar sua nova classe da classe existente. Ao fazer isso, você pode reutilizar os campos e métodos da classe existente, sem ter que escrever e depurar sozinho. (Inheritance - Java Tutorials, 2012)

- **Interface:** Na linguagem de programação Java, uma Interface é um tipo de referência, similar a uma classe, que pode conter apenas constantes, assinaturas de métodos e tipos aninhados. Não há corpos de método. Interfaces não podem ser instanciadas, elas só podem ser implementadas por classes ou estendido por outras interfaces.

Interfaces tem outro papel muito importante na linguagem de programação Java. Interfaces não fazem parte da hierarquia de classes, apesar de trabalhar em conjunto com classes. A linguagem de programação Java não permite herança múltipla, mas as Interfaces fornecem uma alternativa.

Em Java, uma classe pode herdar de apenas uma classe, mas pode implementar mais de uma interface. Portanto, os objetos podem ter vários tipos: o tipo de sua própria classe e os tipos de todas as interfaces que elas implementam. Isto significa que se uma variável é declarada para ser o tipo de uma interface, o seu valor pode fazer referência a qualquer objeto que é instanciado a partir de qualquer classe que implementa a interface. (Interfaces, Java Tutorials, 2012)

4.3.2 InterfaceGraficaDPA (Componentes)

A biblioteca é composta por 38 componentes listados abaixo, acompanha documentação nos padrões JavaDoc e o código fonte é aberto e pode ser alterada respeitando os créditos do autor original.

8 Botões (JButton)

- ButtonAjudaDPA
- ButtonAtualizarDPA
- ButtonBuscarDPA
- ButtonExcluirDPA
- ButtonImprimirDPA
- ButtonLimparDPA
- ButtonSalvarDPA
- ButtonSimplesDPA

3 Combobox (JCombobox)

- ComboboxAutoCompleteDPA
- ComboboxDPA
- ComboboxEstadosBrasilDPA

8 Ícones (JLabel)

- IconAdminDPA
- IconCadeadoDPA
- IconComputadorDPA
- IconEmailDPA
- IconFoneDPA
- IconPastaDPA
- IconUserDPA
- IconUserGroupDPA

7 Campos Formatados (JFormattedTextField)

- JFormattedCepDPA
- JFormattedCnpjDPA
- JFormattedCpfDPA
- JFormattedDataDPA
- JFormattedHoraMinutoDPA
- JFormattedHoraMinutoSegundoDPA
- JFormattedTelefoneDPA

4 Rótulos (JLabel)

- LabelDPA
- LabelMensagemAzulDPA
- LabelMensagemVermelholDPA
- LabelTituloDPA

2 Painéis (JPanel)

- PanelDPA
- PanelTituloDPA

1 Campo Senha (JPasswordField)

- PasswordFieldDPA

1 Área de Texto (JTextArea)

- TextAreaDPA

4 Campos de Texto (JTextField)

- TextFieldDPA
- TextFieldDataHoraDPA
- TextFieldMoedaReal
- TextFieldNotNullDPA

A seguir serão exibidas as telas com os previews dos componentes disponibilizados pela biblioteca.

Figura 27 traz os botões personalizados da biblioteca.

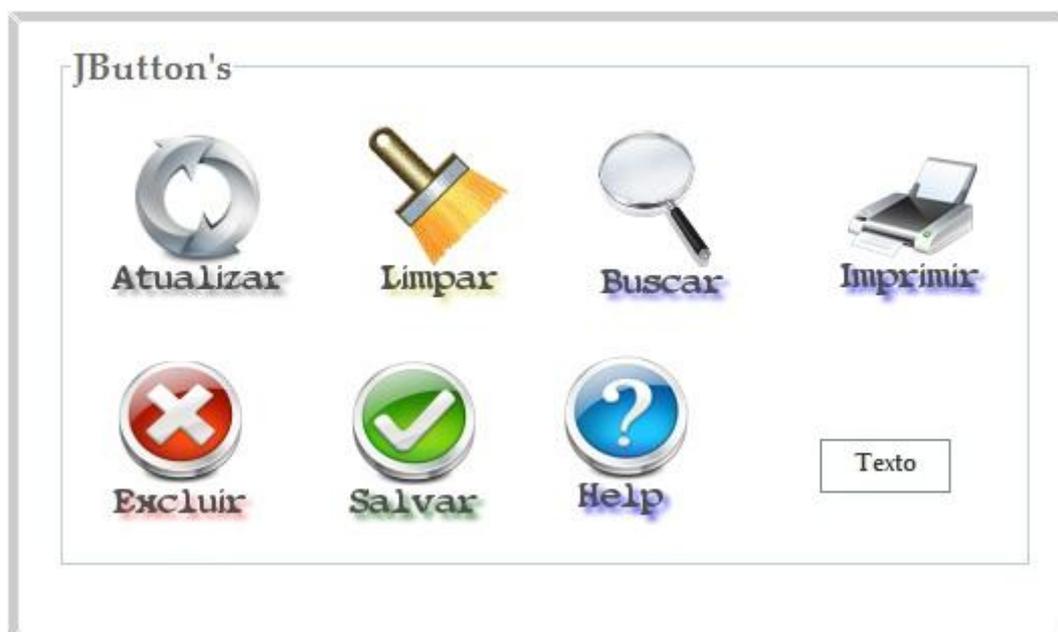


Figura 27 - Botões Personalizados

Figura 28 traz as combobox personalizadas da biblioteca.



Figura 28 - Combobox Personalizados

Figura 29 traz os formattedTextField personalizados da biblioteca.

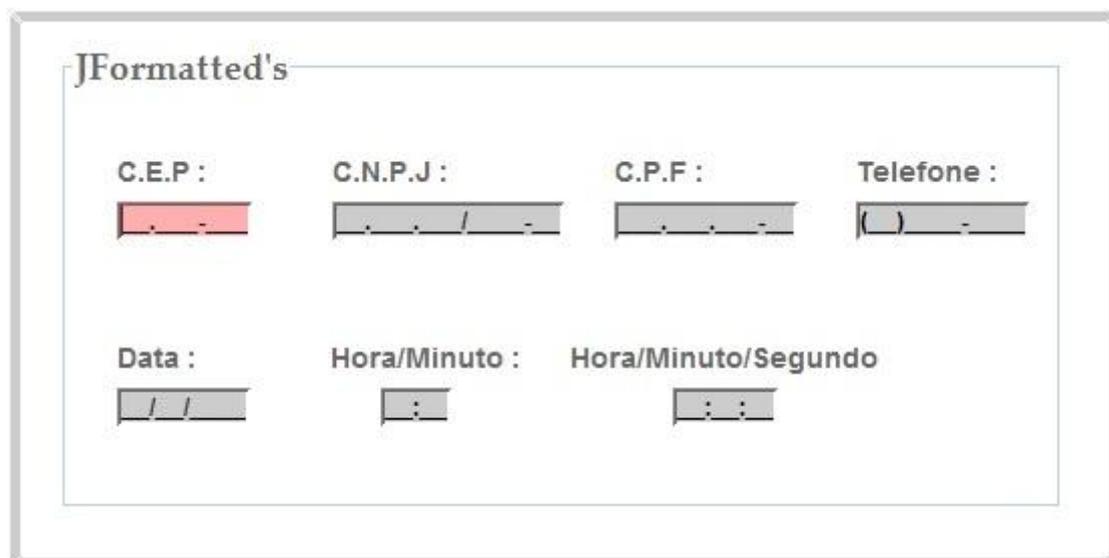


Figura 29 - FormattedTextField Personalizados

Figura 30 traz os Labels (Ícones) personalizados da biblioteca.



Figura 30 - Labels (Ícones) personalizados

Figura 31 traz os Labels personalizados da biblioteca.



Figura 31 - Labels personalizados

Figura 32 traz os Panels personalizados da biblioteca.

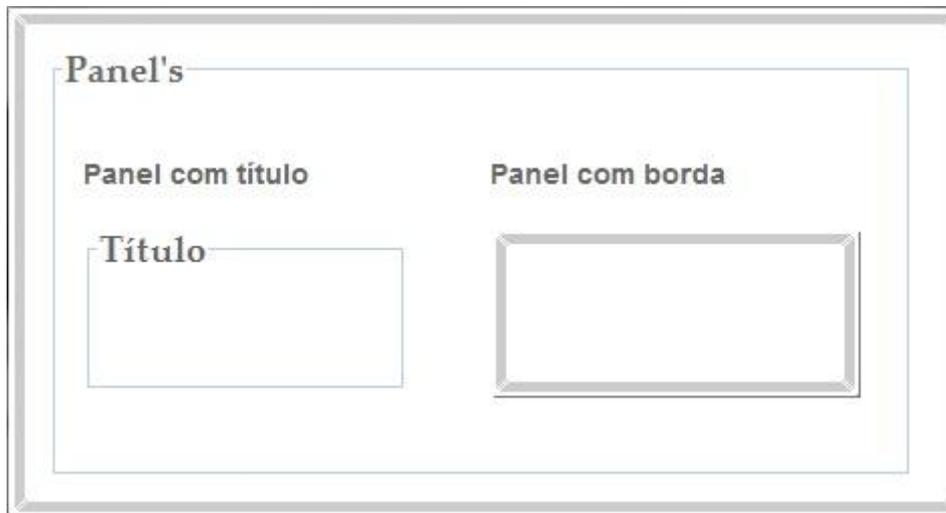


Figura 32 - Panels personalizados

Figura 33 traz o PasswordField personalizado da biblioteca.

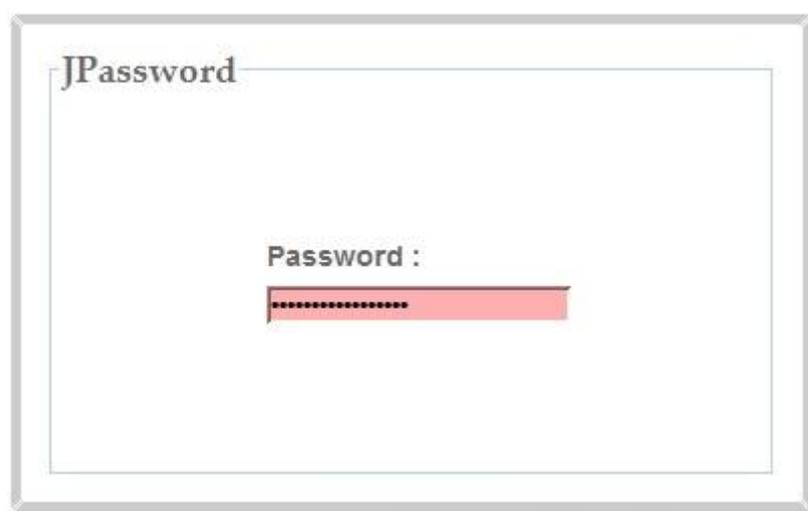


Figura 33 - Password personalizado

Figura 34 traz a TextArea personalizada da biblioteca.



Figura 34 - TextArea personalizado

Figura 35 traz os TextFields personalizados da biblioteca.



Figura 35 - TextField personalizado

4.4 INTERFACEGRAFICADPA (APLICATIVO)

Para apresentar a biblioteca InterfaceGraficaDPA, gerenciar a pré-visualização dos componentes, dos formulários de exemplo e dicas e apresentar o questionário foi desenvolvido um aplicativo denominado InterfaceGraficaDPAPreview.

Esse aplicativo facilita a navegação do usuário por todo conteúdo proposto por esse trabalho. A figura 36 exibe a tela principal do aplicativo InterfaceGraficaDPAPreview.



Figura 36 - Menu Principal (InterfaceGraficaDPAPreview)

As seguintes opções estão disponíveis no menu:

- **InterfaceDPA:** fornece informações sobre a biblioteca Swing e a visualização do trabalho completo em arquivo no formato PDF.
- **Avalie sua Interface:** Fornece o questionário para avaliação de itens básicos para uma boa Interface.
- **Componentes:** oferece um preview de todos os componentes da biblioteca InterfaceGraficaDPA.
- **Exemplos:** exibe os formulários com exemplos e dicas.
- **Sair:** finaliza o aplicativo.

É possível fazer a navegação pelo menu através do mouse ou teclas direcionais no teclado, seguido da tecla Enter para acionar as funções.

4.5 INTERFACEGRAFICADPA (QUESTIONÁRIO)

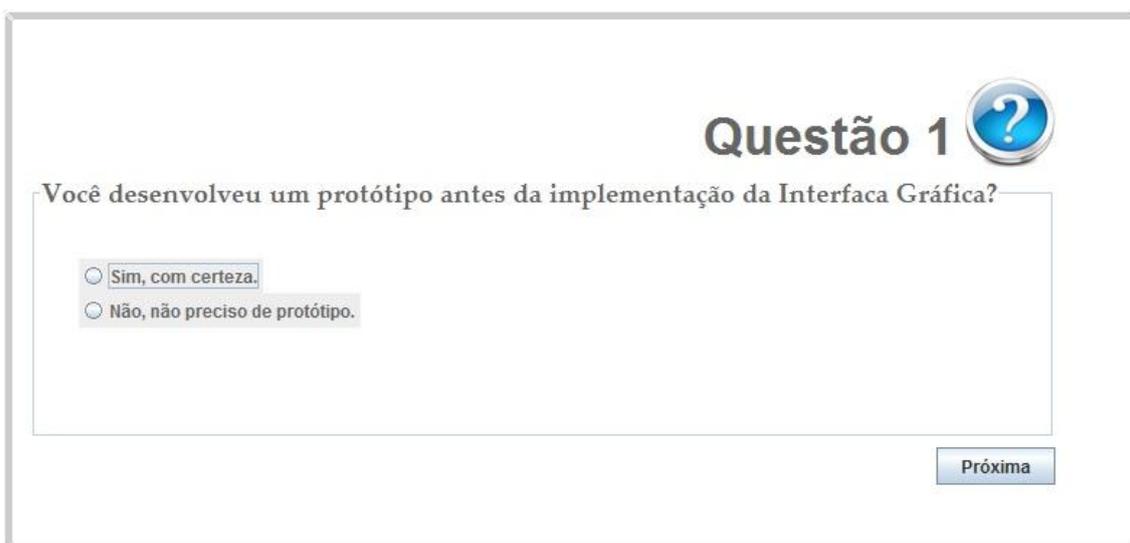
Com o intuito de se fazer uma breve análise de uma Interface Gráfica finalizada ou ainda em construção foi elaborado um questionário baseado nos estudos, que avalia de forma básica os itens essenciais para que uma Interface apresente as configurações mínimas para um bom uso.

O questionário é feito via software, composto por 10 questões de múltipla escolha, onde a pontuação para cada resposta correta é de 10 pontos e para resposta incorreta 0 pontos, sendo então a nota máxima para o resultado final 100 pontos.

O questionário tem também o objetivo de despertar no desenvolvedor uma maior preocupação em relação a fatores que facilitam em muito a vida do usuário final, buscando assim novos conhecimentos e novas maneiras de fornecer ao usuário uma interface mais amigável.

A seguir serão exibidos os snapshots de das questões que compõem o teste.

A figura 37 exibe a primeira questão do Teste.



Questão 1 

Você desenvolveu um protótipo antes da implementação da Interfaca Gráfica?

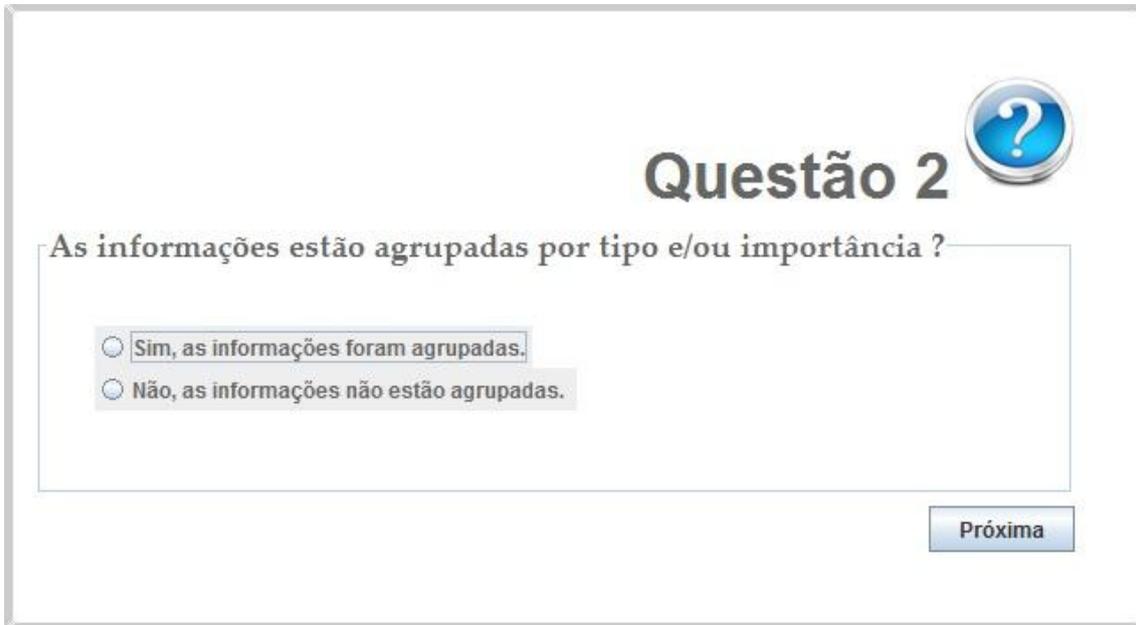
Sim, com certeza.

Não, não preciso de protótipo.

Próxima

Figura 37 - Questão 1 Protótipo

A figura 38 exibe a segunda questão do Teste.



Questão 2 

As informações estão agrupadas por tipo e/ou importância ?

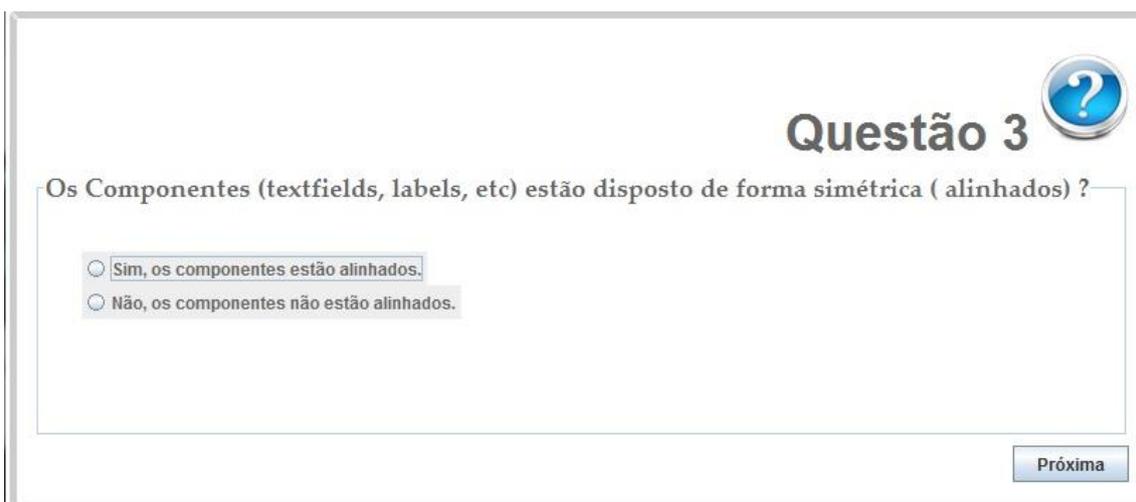
Sim, as informações foram agrupadas.

Não, as informações não estão agrupadas.

Próxima

Figura 38 - Questão 2 Agrupamento

A figura 39 exibe a terceira questão do Teste.



Questão 3 

Os Componentes (textfield, labels, etc) estão disposto de forma simétrica (alinhados) ?

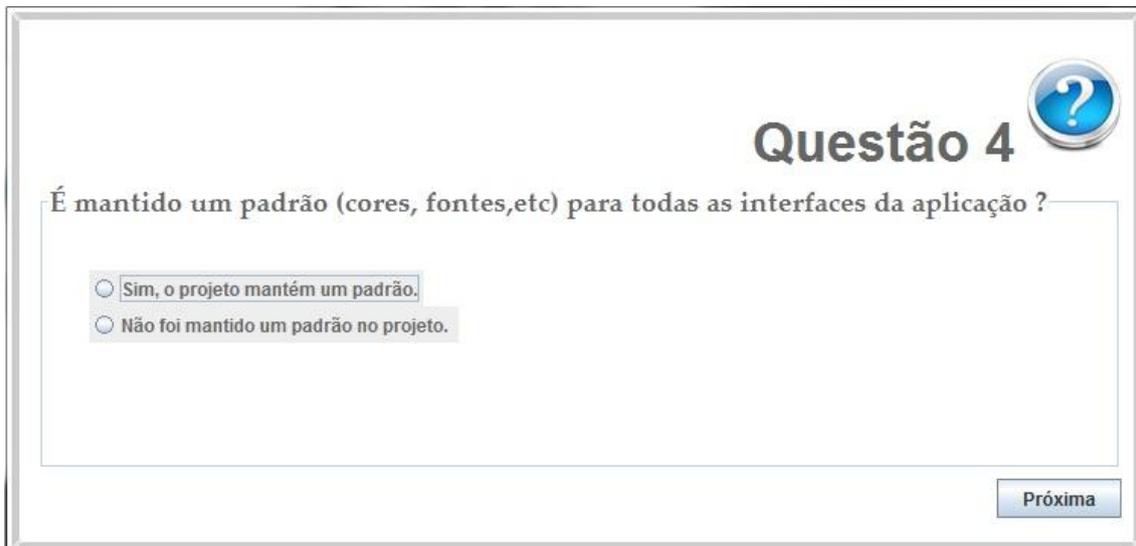
Sim, os componentes estão alinhados.

Não, os componentes não estão alinhados.

Próxima

Figura 39 - Questão 3 Simetria

A figura 40 exibe a quarta questão do Teste.



The screenshot shows a test question interface. At the top right, it says "Questão 4" next to a blue question mark icon. The question text is "É mantido um padrão (cores, fontes,etc) para todas as interfaces da aplicação?". Below the question, there are two radio button options: "Sim, o projeto mantém um padrão." and "Não foi mantido um padrão no projeto.". At the bottom right, there is a "Próxima" button.

Figura 40 - Questão 4 Padrão entre todas Interfaces

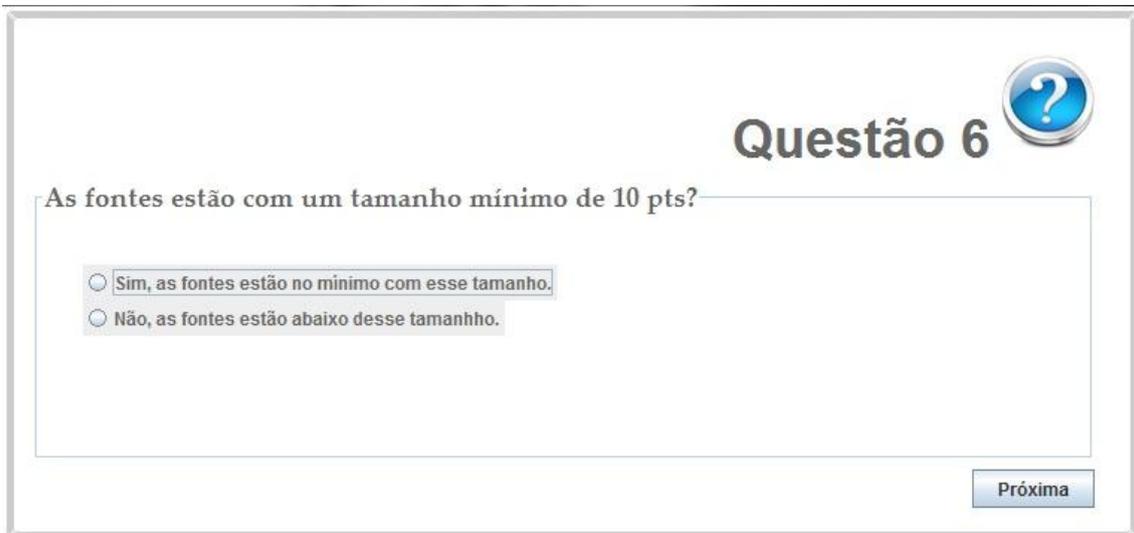
A figura 41 exibe a quinta questão do Teste.



The screenshot shows a test question interface. At the top right, it says "Questão 5" next to a blue question mark icon. The question text is "As cores do background (plano de fundo) estão entre os tons de cinza claro ou branco?". Below the question, there are three radio button options: "Sim, estão entre essas cores.", "Não, utilizei outras cores.", and "Não, eu utilizo uma imagem como plano de fundo.". At the bottom right, there is a "Próxima" button.

Figura 41 - Questão 5 Cor (Background)

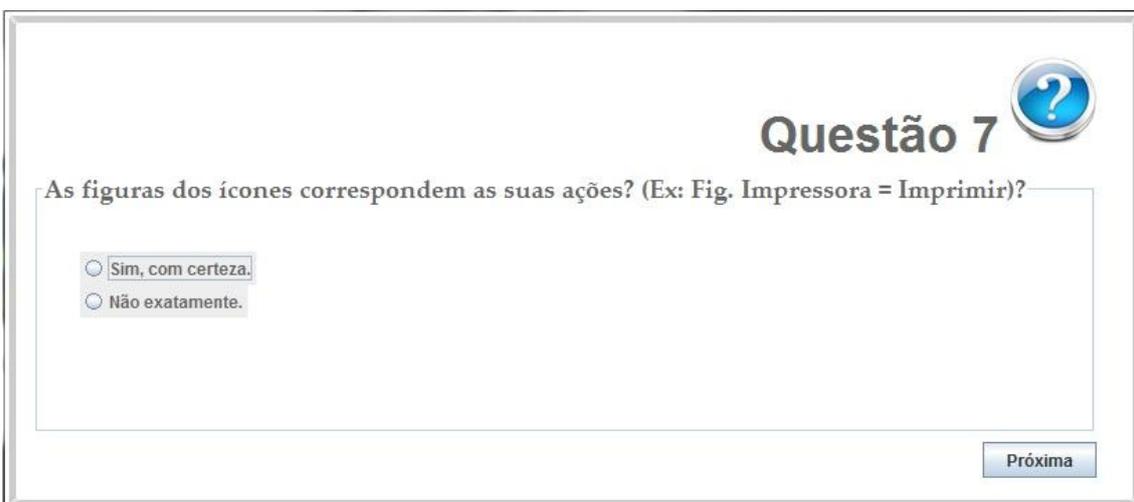
A figura 42 exibe a sexta questão do Teste.



The screenshot shows a test question interface. At the top right, it says "Questão 6" next to a blue circular icon with a white question mark. Below this, the question text is "As fontes estão com um tamanho mínimo de 10 pts?". There are two radio button options: "Sim, as fontes estão no mínimo com esse tamanho." and "Não, as fontes estão abaixo desse tamanho.". At the bottom right, there is a blue button labeled "Próxima".

Figura 42 - Questão 6 Fonte (Tamanho Mínimo)

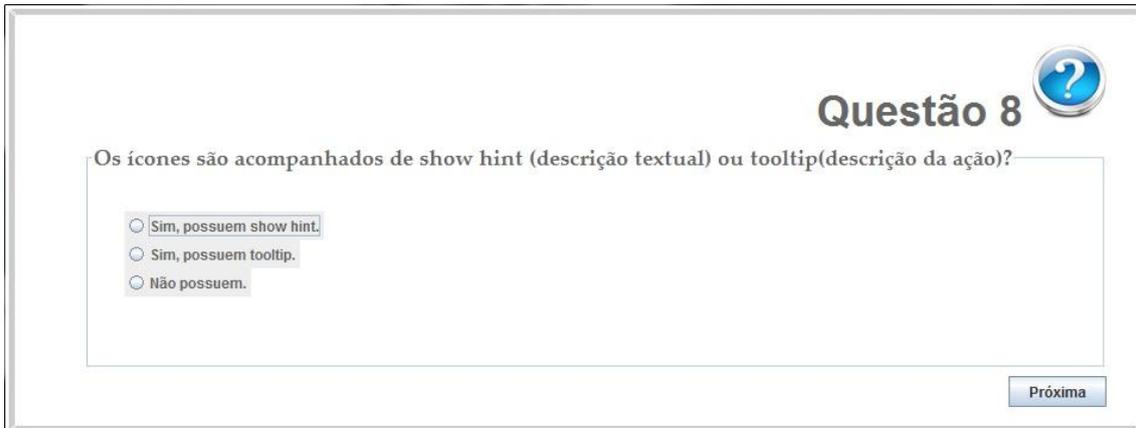
A figura 43 exibe a sétima questão do Teste.



The screenshot shows a test question interface. At the top right, it says "Questão 7" next to a blue circular icon with a white question mark. Below this, the question text is "As figuras dos ícones correspondem as suas ações? (Ex: Fig. Impressora = Imprimir)?". There are two radio button options: "Sim, com certeza." and "Não exatamente.". At the bottom right, there is a blue button labeled "Próxima".

Figura 43 - Questão 7 Ícones (Figuras)

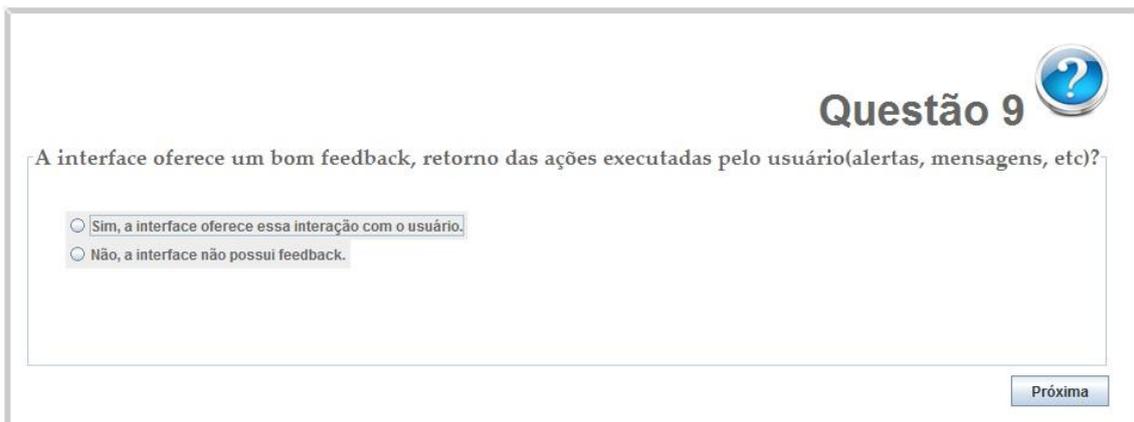
A figura 44 exibe a oitava questão do Teste.



The screenshot shows a test question interface. In the top right corner, there is a blue circular icon with a white question mark. Below it, the text "Questão 8" is displayed in a bold, black font. The main question text is "Os ícones são acompanhados de show hint (descrição textual) ou tooltip(descrição da ação)?". Below the question, there are three radio button options: "Sim, possuem show hint.", "Sim, possuem tooltip.", and "Não possuem.". The first option is selected. In the bottom right corner, there is a blue button labeled "Próxima".

Figura 44 - Questão 8 Ícones (Legenda)

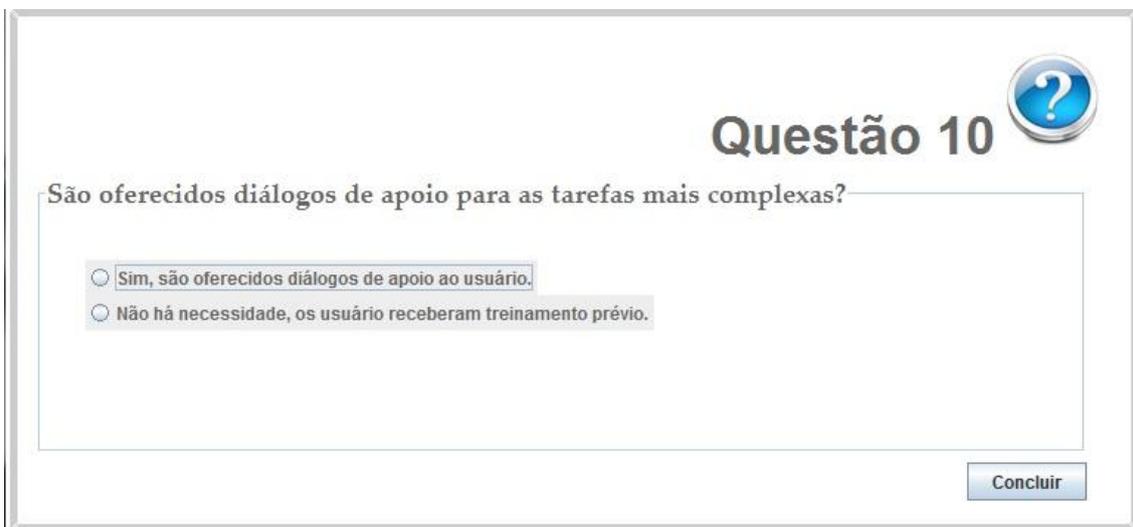
A figura 45 exibe a nona questão do Teste.



The screenshot shows a test question interface. In the top right corner, there is a blue circular icon with a white question mark. Below it, the text "Questão 9" is displayed in a bold, black font. The main question text is "A interface oferece um bom feedback, retorno das ações executadas pelo usuário(alertas, mensagens, etc)?". Below the question, there are two radio button options: "Sim, a interface oferece essa interação com o usuário." and "Não, a interface não possui feedback.". The first option is selected. In the bottom right corner, there is a blue button labeled "Próxima".

Figura 45 - Questão 9 Mensagens de alerta

A figura 46 exibe a décima e última questão do Teste.



Questão 10 

São oferecidos diálogos de apoio para as tarefas mais complexas?

Sim, são oferecidos diálogos de apoio ao usuário.

Não há necessidade, os usuário receberam treinamento prévio.

Concluir

Figura 46 - Questão 10 Help (Ajuda)

A figura 47 exibe o resultado final para um teste que obteve pontuação máxima.



 **Resultado**

Avaliação Respostas (Dicas)

Questão 1 - Protótipo OK!

Questão 2 - Agrupamento OK!

Questão 3 - Alinhamento e Simetria OK!

Questão 4 - Padrão nas Interfaces OK!

Questão 5 - Cores Background OK!

Questão 6 - Fontes Tamanho Mínimo OK!

Questão 7 - ícones e suas figuras OK!

Questão 8 - ícone com show hint OK!

Questão 9 - Feedback OK!

Questão 10 - Diálogos de apoio (Help) OK!

Nota Final :100

Sair

Figura 47 - Resultado de um teste com 100 pontos

4.6 PROTOTIPAÇÃO

A prototipação é fundamental para o desenvolvimento de uma boa interface gráfica. Através dela pode-se prever falhas, detectar a falta de algum serviço e obter um bom “feedback” do usuário antes da implementação final.

Segundo Nielsen (1993) a ideia por trás da prototipação é de diminuir a complexidade da implementação pela eliminação de partes do sistema. Os cenários podem ser implementados como maquetes em papel ou em ambientes de prototipação evoluídos.

O objetivo dos protótipos podem ser divididos em:

- **Horizontal:** Exibe a interface do usuário sem ter o foco nas funcionalidades por trás dos botões, demonstrando superficialmente toda a interface. Este tipo de protótipo permite testar a interface como um todo.
- **Vertical:** Tem seu foco nas funcionalidades do sistema. Possui poucas tarefas, mas funcionalmente aprofundadas. Este tipo de protótipo permite testar uma pequena parte do sistema.

A figura 48 traz as fases do processo de prototipação.



Figura 48 - Fases do Processo de Prototipação

- **Definição dos objetivos do protótipo:** É feito um estudo das regras de negócio do sistema que deverá ser criado. O objetivo é verificar a viabilidade da prototipação, o material a ser utilizado.
- **Definição das funcionalidades do protótipo:** Este é o momento em que são combinadas e registradas quais são as funcionalidades mais desejadas e mais importantes que devem estar presentes.
- **Desenvolvimento do protótipo:** Nesta fase é construído o protótipo de acordo com as especificações geradas nas fases anteriores.
- **Entrega e avaliação do protótipo:** O final de uma iteração no desenvolvimento de um protótipo é a entrega do mesmo, podendo este ser demonstrado aos usuários, extraindo a opinião dos mesmos a respeito do protótipo, verificando assim, as possibilidades de modificações e novas ideias sobre o objeto final a ser entregue.

4.6.1 Prototipação e Formulários de Exemplos

Nesta seção serão exibidos snapshots dos protótipos para os formulários de exemplos desenvolvidos para o material de apoio, os protótipos aqui apresentados foram desenvolvidos na ferramenta Microsoft Visio (versão free para estudantes). Existem várias ferramentas gratuitas que possibilitam o desenvolvimento de protótipos com excelente qualidade.

O objetivo é mostrar que com a utilização dos protótipos, a implementação se torna muito mais organizada, fica mais fácil a detecção de falhas e pode-se obter uma aprovação do usuário antes do desenvolvimento em si.

A figura 49 traz um “print” de um formulário de Cadastro de Dados Pessoais e Dados de Endereço, onde somente as posições e agrupamentos foram definidos. Fica nítido a importância da prototipação para uma melhor organização da Interface que se objetiva fazer.

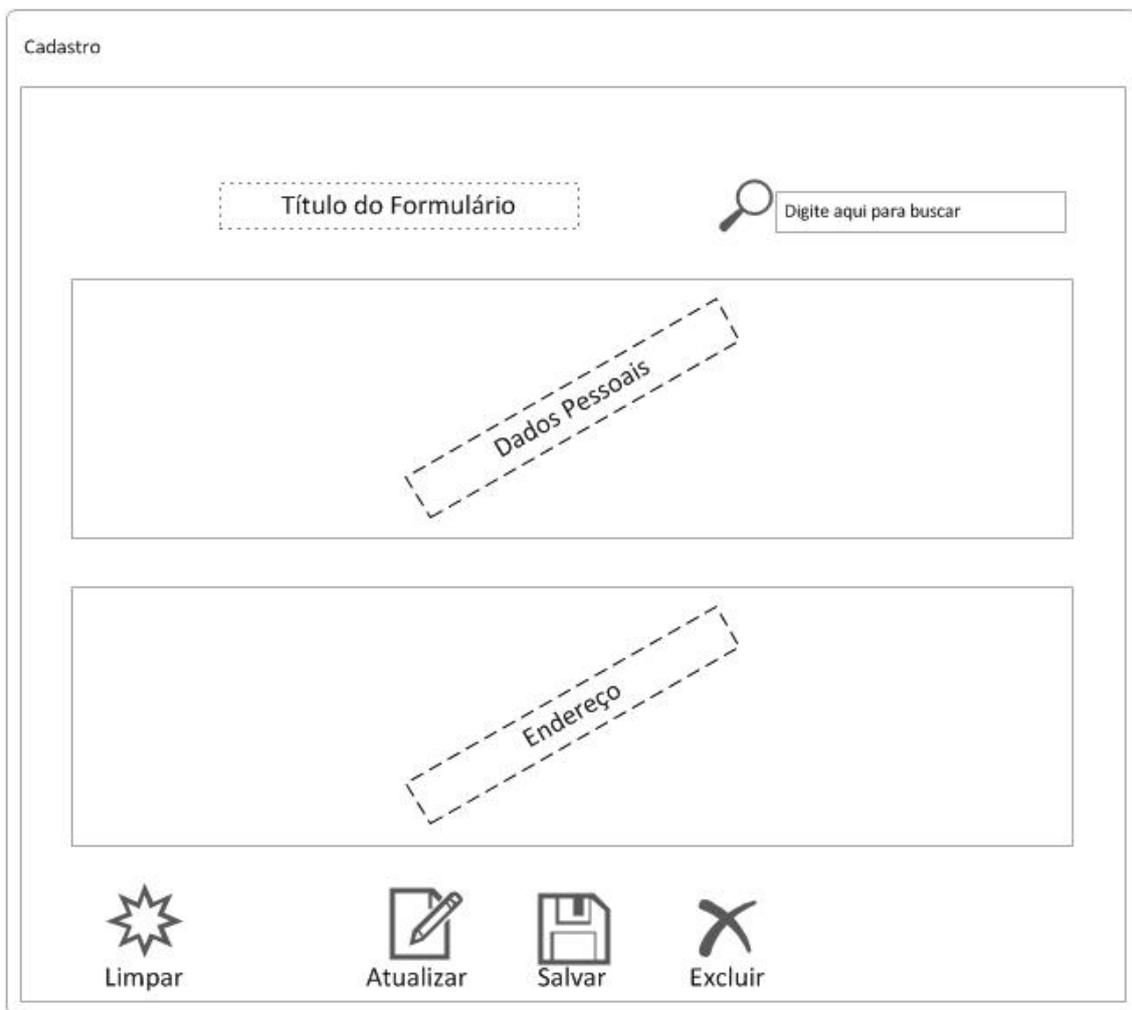


Figura 49 - Protótipo Interface Cadastro Dados Pessoais e Endereço

Na figura 50 pode-se observar a Interface implementada com base no protótipo da figura 49. A interface final nem sempre ficara totalmente fiel ao protótipo original, mais fornece ao desenvolvedor uma boa referência e se possível, aprovada pelo usuário final.

Alguns desenvolvedores preferem utilizar a própria IDE de desenvolvimento como ferramenta de protótipo, mas vale ressaltar que nesse caso código fonte está sendo gerado.

Cadastro

Consultar cadastrado:  **Buscar**

Dados Pessoais

Nome: Status:

Estado civil: Sexo: Nascimento: C.P.F.: Escolaridade:

Endereço

Rua: Número: Apto:

Bairro: Cidade: U.F.: C.E.P.:

 **Limpar**  **Excluir**  **Atualizar**  **Salvar**

Figura 50 - Interface de exemplo Implementada com base em protótipo

Na figura 51 tem-se um protótipo de um exemplo clássico de Login de Sistema. Nesse caso todos os campos já foram pré-definidos, assim como o agrupamento das informações, os avisos pertinentes e também foi reservado uma área para a possível colocação de um ícone.

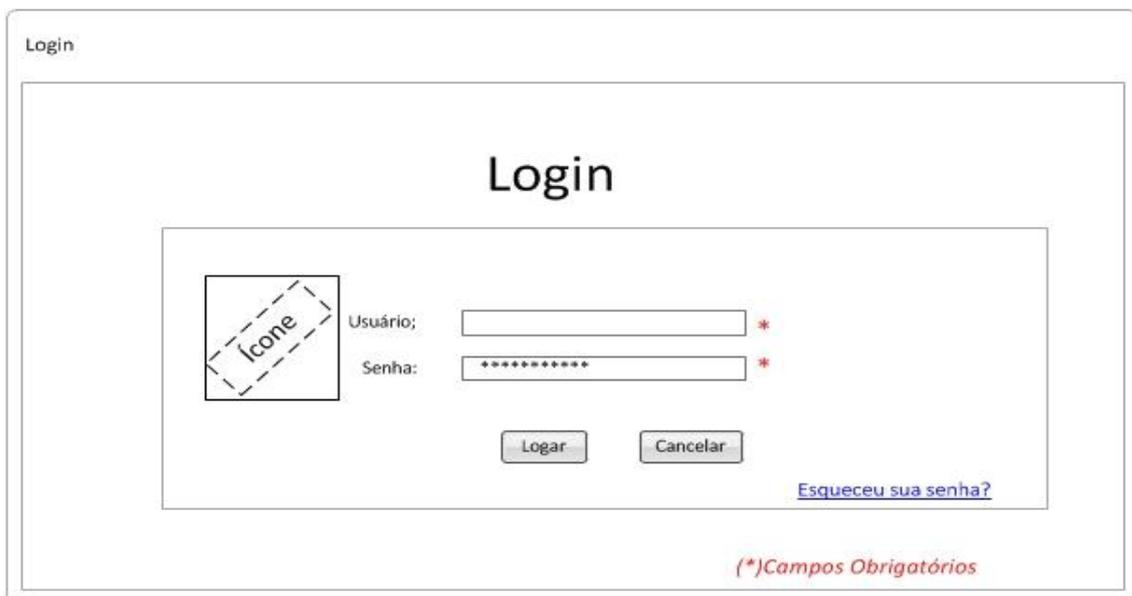


Figura 51 - Protótipo de uma tela de Login

A figura 52 exibe um "snapshot" da tela de Login devidamente implementada, nesse caso, sendo o mais fiel possível ao protótipo, respeitando todos os itens que foram sugeridos. O campo em destaque com o nome do usuário, é que recebe o foco ao inicializar da tela.

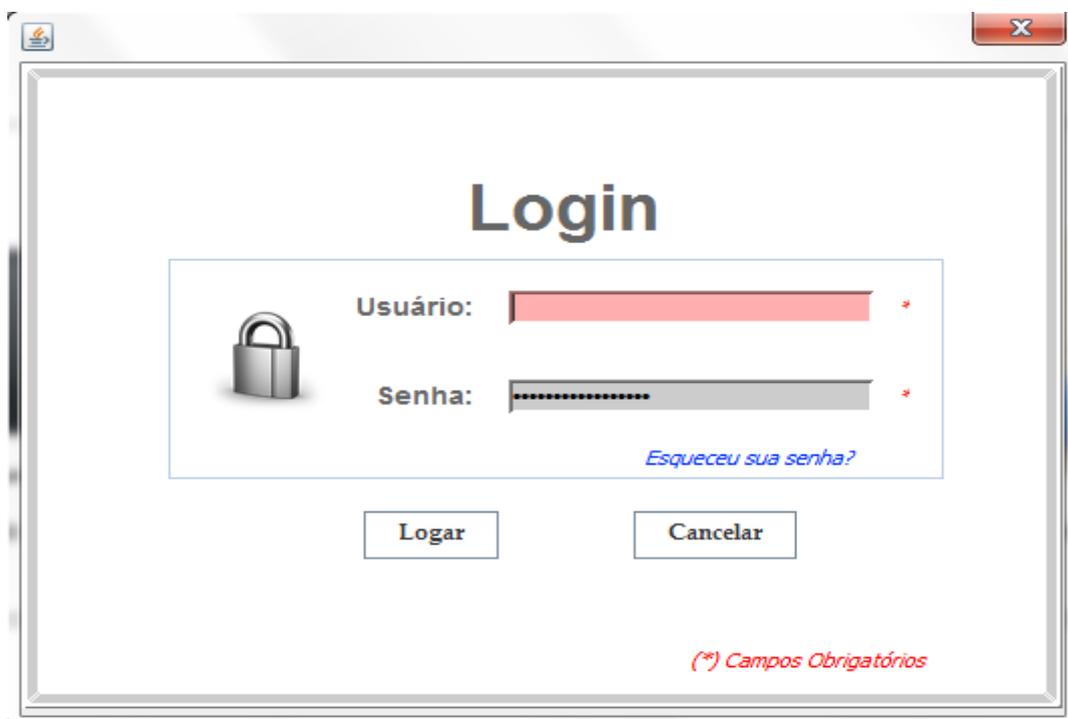


Figura 52 - Interface de exemplo (Login) implementada com base em protótipo

A figura 53 traz mais um protótipo trivial em sistemas, o cadastro de usuários, como no exemplo anterior, esse protótipo já disponibiliza um exemplo completo sugerindo todos os campos, botões e avisos pertinentes a esse tipo de tela, as informações são devidamente agrupadas para evitar a ideia de componentes soltos na tela e um espaço para um possível ícone também é reservado.

Cadastro de Usuário

Cadastro de Usuário

Ícone

Usuário: *

Senha: *

Conf. Senha: *

Tipo Usuário:

(*) Campos Obrigatórios

Figura 53 - Protótipo de tela para cadastro de usuários de sistema

Na figura 54 pode-se observar a Interface de cadastro de usuário devidamente implementada respeitando o que foi estipulado no protótipo. Alguns desenvolvedores questionam o fato de que se implementaria duas vezes a Interface devido a prototipação, mas o fato é que a partir do protótipo o desenvolvedor chega na fase de implementação das telas com a ideia formada, quantidade de campos e componentes definidos, aprovação do usuário, sem geração de código-fonte desnecessário.

The image shows a software window titled "Cadastro de Usuários" (User Registration). On the left side of the window is a small icon of a person. The main area contains a form with the following fields:

- Usuário:** A text input field with a red asterisk to its right, indicating it is mandatory.
- Senha:** A password input field with a grey background and a red asterisk to its right.
- Conf. Senha:** A confirmation password input field with a grey background and a red asterisk to its right.
- Tipo Usuário:** A dropdown menu currently showing "Admin".

At the bottom of the form are two buttons: "Cadastrar" (Register) and "Cancelar" (Cancel). Below the form, there is a red note: "(*) Campos Obrigatórios" (Mandatory fields).

Figura 54 - Interface de exemplo(Cadastro de Usuários) implementada com base em protótipo

A figura 55 mostra um exemplo simples de um protótipo de um Menu para uma aplicação utilizando apenas ícones e o show hint para os mesmos.

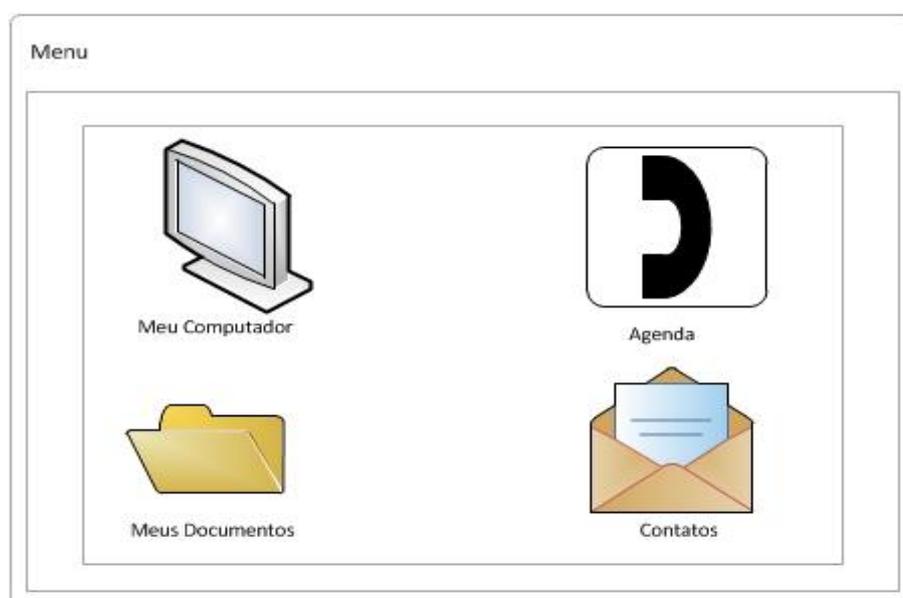


Figura 55 - Protótipo de Menu utilizando somente ícones com show hint

Na figura 56 pode-se observar a Interface de Menu devidamente implementada baseada no protótipo apresentado na figura 55, a fidelidade ao protótipo é mantida durante o processo de implementação.

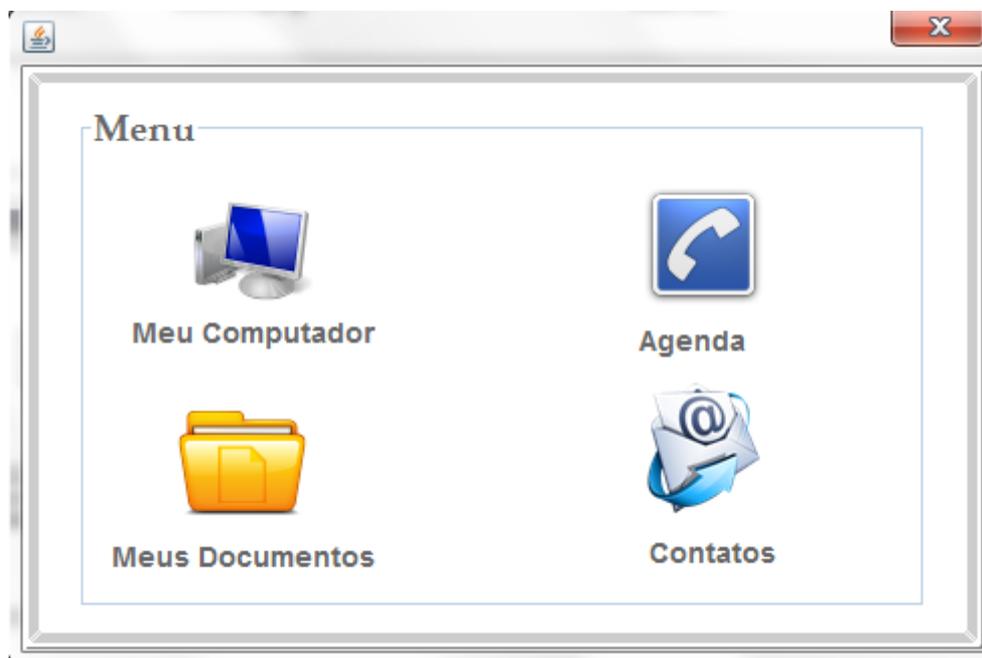


Figura 56 - Interface de Exemplo (Menu) implementada com base em protótipo

A prototipação concebe ao desenvolvedor uma ótima base para a implementação da Interface Gráfica. Fazendo uma comparação é como se fosse a maquete para o engenheiro, onde é possível se obter uma visão bem próxima do que se deseja obter durante o processo de implementação. As técnicas de prototipação são diversas e válidas para todo tipo de desenvolvimento de software. Alguns desenvolvedores utilizam prototipação em papel entre outros, o importante é que no mínimo se obtenha um esboço da interface que se deseja apresentar e que se possível a mesma seja aprovada pelo usuário final, mantendo-se as ressalvas necessárias.

4.6.2 Formulário (Dicas)

Assim como no capítulo 3 desse trabalho na seção 3.6 Estudos de Caso (Interface Falhas existentes no mercado), foram apontando alguns erros cruciais no desenvolvimento de Interfaces Gráficas, para esse material de apoio foi desenvolvida uma Interface que possui ícones com algumas dicas.

A figura 57 traz uma Interface de exemplo ao clicar no ícone uma mensagem com a dica correspondente é exibida.

The image shows a web interface for a registration system. At the top left, there is a large 'Cadastro' title with an information icon. To the right, there is a search bar labeled 'Consultar cadastrado:' with a 'Buscar' button. Below this, the form is divided into two main sections: 'Dados Pessoais' and 'Endereço'. The 'Dados Pessoais' section includes fields for 'Nome:', 'Estado civil:', 'Sexo:', 'Nascimento:', 'C.P.F.', and 'Escolaridade:', along with a 'Status:' dropdown menu. The 'Endereço' section includes fields for 'Rua:', 'Número:', 'Apto:', 'Bairro:', 'Cidade:', 'U.F.', and 'C.E.P.'. At the bottom, there are three buttons: 'Limpar' (with a brush icon), 'Excluir' (with a red 'X' icon), 'Atualizar' (with a circular refresh icon), and 'Salvar' (with a green checkmark icon). Blue circles and arrows are overlaid on the image, pointing to various elements: the search bar, the 'Nome:' field, the 'Status:' dropdown, the 'Sexo:' dropdown, the 'Nascimento:' field, the 'C.P.F.' field, the 'Endereço' section, the 'Limpar' button, and the 'Excluir', 'Atualizar', and 'Salvar' buttons.

Figura 57 - Interface de Exemplo (Ícones com dicas)

5 Conclusão

Esse trabalho que teve como o foco a Interface Gráfica durante o processo de desenvolvimento de software, foi motivado pela anseio de facilitar o dia a dia dos usuários de sistemas computacionais e promover ajuda e incentivo aos desenvolvedores de software em geral.

5.1 CONSIDERAÇÕES FINAIS

As principais contribuições desse trabalho são o material de apoio e estudos que acompanham o mesmo em anexo, as pesquisas realizadas sobre o tema proposto, além é claro da iniciativa tomada de se valorizar mais a interface gráfica nos diferente tipos de desenvolvimento (Desktop, Web, Mobile) e plataformas.

É fato que pode-se estender o tema, ou restringir-se a áreas de interesse para estudos mais profundos e precisos. Os estudos realizados para esse trabalho, focam em fatores simples e erros corriqueiros que prejudicam o desempenho de interfaces.

Foram feitos durante o decorrer do trabalho estudos que serviram de base para o apontamento de erros, para sugestões e dicas e para construção de uma biblioteca de componentes e um questionário básico. Sempre que possível, para maior clareza foram exibidas tabelas e imagens para facilitar o entendimento do assunto.

O material de apoio resultante foi testado e mostrou se eficaz para o que se propõem, a biblioteca InterfaceGraficaDPA foi desenvolvida e documentada de maneira que novos componentes possam ser adicionados futuramente.

Para disponibilizar todo conteúdo deste trabalho foi criado um website (<http://www.interfacegraficadpa.weebly.com>), na aba download todo conteúdo (aplicativo, biblioteca, código-fonte, javadoc) pode ser baixado e conferido.

A figura mostra a página inicial do site que disponibiliza o conteúdo deste trabalho online.



Figura 58 - WebSite InterfaceGraficaDPA

5.2 TRABALHOS FUTUROS

Tomando como base esse trabalho, ideias semelhantes podem surgir, ou a proposta pode ser continuada, materiais de avaliação, correção, novas bibliotecas para desenvolvimento Desktop, Web ou Mobile podem ser construídas e disponibilizadas.

Outra sugestão ainda mais inovadora seria a construção de um software com a capacidade de analisar os itens de uma interface gráfica através da análise do código fonte, baseando-se em um dicionário ativo de parâmetros.

REFERÊNCIAS

ABNT – Associação Brasileira de Normas Técnicas, **ISO NBR 9241-Requisitos ergonômicos para trabalho de escritório com computadores Parte 10 - Princípios de diálogo**, 2000.

ABNT – Associação Brasileira de Normas Técnicas, **ISO NBR 9241-11 Requisitos Ergonômicos para Trabalho de Escritórios com Computadores Parte 11 – Orientações sobre Usabilidade**, 2000.

ABNT – Associação Brasileira de Normas Técnicas, **NBR ISO/IEC 9126-1 Engenharia de software - Qualidade de produto Parte 1: Modelo de qualidade**, 2003.

BATTISTELLA M. R., **A importância da cor em ambientes de trabalho um estudo de caso**, UFSC (Universidade Federal de Santa Catarina), 2003.

BELTRÁN, B., **An EPSS interface that people can use**. Disponível em: <http://ourworld.compuserve.com/homepages/bea_beltran/>. Acesso em Janeiro de 2012.

BRANDÃO, E. R., **Definições de design, ergonomia usabilidade, interação humano-computador e ergodesign**, PUC-Rio, 2006.

BRANDÃO, E. R., **Publicidade on-line, ergonomia e usabilidade: o efeito de seis tipos de banner no processo humano de visualização do formato do anúncio na tela do computador e lembrança da sua mensagem**, PUC-Rio, 2006.

Computer Hope., **GUI**. Disponível em: <<http://www.computerhope.com/jargon/g/gui.htm>> Acesso em Janeiro de 2012. Cotidiano, 2004.

CYBIS, W.; BETIOL, A. H. e FAUST. R., **Ergonomia e Usabilidade Conhecimentos, Métodos e Aplicações**, Novatec, 2007.

DA ROCHA, H. V.; BARANAUSKAS, M. C. C., **Design e avaliação de interfaces humano-computador**. São Paulo, IME-USP, 2000.

DE OLIVEIRA. F. B., **Interfaces usuário-máquina**, Unimontes, 2007.

ENDLER, A. M. e PIMENTA, M. S., **Inserindo IHC em Empresas Brasileiras de Informática: Relato, Discussão e Lições Aprendidas**, VI Simpósio sobre

Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano, UFPR, 2004.

FERNANDES, G. G., **Interface Humano Computador:prática pedagógica para ambientes virtuais**, UFPI (Universidade Federal do Piauí) , 2009.

FERNANDES, J. H. C., **As 10 Áreas da Engenharia de Software, Conforme o SWEBOK**, 2004.

FERREIRA, A. B. H., **Minidicionário Aurélio, da Língua Portuguesa, 7ª Edição**, Curitiba, Editora Positivo Ltda, 2004.

FONTANA. L. A., **Estudo da cor**, UCS (Universidade Católica de Goiás) , 2012.

GALVIS, A. H. P., **Engenharia de Software**, Colômbia, Adicione Uniandes, 1992.

Hyman, W. A., **Errors in The Use of Medical Equipment**. Human Error in Medicine. Lawrence Erlbaum Associates, Inc. London, 1994.

IEEE STD 610.12-1990, **IEEE Standard Glossary of Software Engineering Terminology**, IEEE Computer Society, 1990.

Java.com., **O que é a tecnologia Java e por que é necessária?** .Disponível em:<http://www.java.com/pt_BR/download/faq/whatis_java.xml?printFriendly=true> Acesso em Julho de 2012.

Java.com., **Inheritance**. Disponível em:<<http://docs.oracle.com/javase/tutorial/java/landl/subclasses.html> > Acesso em Agosto de 2012.

Java.com., **Interfaces** .Disponível em:<<http://docs.oracle.com/javase/tutorial/java/landl/createinterface.html> > Acesso em Julho de 2012.

JORDAN, Patrick W., **An Introduction to Usability**, London, UK: Taylor & Francis, 1998.

KOSCIANSKI A e SOARES, M. S., **Qualidade de Software Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software, 2ª Edição**, São Paulo, Novatec, 2007.

MAIA J. F., **Cores em interfaces de softwares - Desenvolvimento de Interfaces**, 2012.

Microsoft., **Uma história do Windows Destaques dos primeiros 25 anos.** Disponível em: < <http://windows.microsoft.com/pt-BR/windows/history>> Acesso em Janeiro de 2012.

Naur, P e B.: Randell., **Software Engineering: a Report on a Conference Sponsored by the NATO Science Committee**, NATO, 1969.

NIELSEN, J., **Usability engineering**. San Francisco: Morgan Kaufman, 1994.

NORMAN, D., **The design of everyday things**, New York, Doubleday, 1988.

NORMAN, K. L., **The design of electronic educational environments: issues and principles in human-computer interaction**, 2003.

PRESSMAN, R., **Engenharia de Software**, Rio de Janeiro, McGraw Hill, 2002.

ROVERI, J. L., **Colorindo com segurança**. Revista proteção, v.5 n.53,p.20-21,1996.

SBC (Sociedade Brasileira de Computação)., **Interação Humano-Computador**. Disponível em:<http://www.sbc.org.br/index.php?option=com_content&view=category&layout=blog&id=45&Itemid=66> Acesso em: Janeiro de 2012.

SILVA, Cassandra Ribeiro de O., **Bases pedagógicas e ergonômicas para concepção e avaliação de produtos educacionais informatizados**. Disponível em: <<http://www.eps.ufsc.br/disserta98/ribeiro/>>. Acesso em: Fevereiro de 2012.

SOMMERVILLE, I; **Engenharia de software. 6. ed.** São Paulo,Addison Wesley, 2003.

SOMMERVILLE, I., **Sommerville Software Engineering, 8th edition**, Pearson Addison-Wesley, 2006

SWEBOK, **Guide to the Software Engineering Body of Knowledge**, IEEE Computer Society, 2004. Disponível em: < <http://swebok.org>>. Acesso em Março de 2012.

TEIXEIRA D. V.; SANTOS I. J. A. L.; FERRAZ, F. T. e CARVALHO, P. V. R., **Ergonomia na concepção de interfaces gráficas de salas de controle avançadas de reatores nucleares**, UTFPR, 2007.