

UNIVERSIDADE DO SUL DE SANTA CATARINA

RAFAEL ORIVALDO LESSA

Glist – Um Checklist Automatizado para Usabilidade

Palhoça(SC)

2006

RAFAEL ORIVALDO LESSA

Glist – Um Checklist Automatizado para Usabilidade

Monografia submetida à Universidade do Sul de Santa Catarina como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação, orientado pela professora Vera Rejane Niedersberg Schuhmacher.

Orientadora Vera Rejane Niedersberg Schuhmacher

**Palhoça(SC)
2006**

RAFAEL ORIVALDO LESSA

Glist – Um Checklist Automatizado para Usabilidade

Monografia submetida à Universidade do Sul de Santa Catarina como requisito parcial à obtenção do título de Bacharel em Sistemas de Informação, orientado pela professora Vera Rejane Niedersberg Schuhmacher

Universidade do Sul de Santa Catarina

Palhoça, de Junho de 2006.

Msc. Vera Rejane Niedersberg Schuhmacher
Universidade do Sul de Santa Catarina

Dr. Mauro Notarnicola Madeira
Universidade do Sul de Santa Catarina

Msc. Flavia Lumi Matuzawa
Universidade do Sul de Santa Catarina

DEDICATÓRIAS

Aos meus pais, meu irmão, que me apoiaram, deram força e foram compreensivos no período em que estive envolvido com este projeto.

AGRADECIMENTOS

Primeiramente a Deus pela oportunidade ímpar de conceber-nos a vida e iluminar-me nesta longínqua caminhada em busca do saber.

Aos meus pais Edson e Sandra por tudo o que fizeram ou deixaram de fazer por minha causa e por ter-me dado à oportunidade de aqui chegar.

À minha querida orientadora, professora Vera R. N. Schuhmacher, por ter aceitado me guiar na definição e condução deste trabalho, na transmissão do seu conhecimento e pela compreensão nos momentos difíceis.

Ao meu irmão Edson Orivaldo Lessa Júnior, pela contribuição através da sua experiência em desenvolvimento de sistemas, e pelo investimento de se tempo em prol deste.

À minha namorada Isabel Cristina Hammes que mesmo estando passando por uma fase difícil de sua vida me compreendeu nos momentos de ausência e pelo apoio, carinho e consolo nas ocasiões de desespero. E aos seus pais, pela força dada e pelas orações em busca de luz e sucesso.

A professora Maria Inés Castiñeira pelas horas de conversas, dicas e conselhos.

Aos meus amigos que, mesmo de longe, sempre estiveram preocupados e interessados em saber como estava o andamento deste.

Aos meus parentes: tias, tios pelo incentivo e interesse em saber do andamento do trabalho.

A todas as pessoas que de alguma forma contribuíram para que mais este degrau de minha vida fosse alcançado.

A todos, o meu muito obrigado!

RESUMO

A usabilidade busca apoiar o processo de interação com o usuário, transmitindo de forma eficaz e eficiente a interação do usuário com a interface de um sistema computacional. Pode ser composta por múltiplos componentes estando tradicionalmente associada a atributos como: facilidade de aprendizagem do sistema; eficiência de uso; capacidade com que o sistema promove a facilidade de memorização das ações necessárias para que o usuário atinja seus objetivos; controle de gestão de erros minimizando e refutando situações possíveis de ocorrências; e principalmente promovendo a satisfação do usuário ao ser utilizado. Com o crescimento da tecnologia a usabilidade começou a ser fonte de preocupação, uma vez que, com um mercado tão dependente da tecnologia precisamos de produtos que sejam fáceis de aprender e utilizar. O presente projeto tem por objetivo o estudo da usabilidade para interfaces e o desenvolvimento de uma ferramenta automatizada de avaliação que ofereça apoio à usabilidade no design de projetos, para desenvolvedores que necessitam de apoio na avaliação de interfaces. A avaliação da usabilidade de um produto de software não é uma garantia absoluta da qualidade do produto, mas é com certeza um fator que em sua ausência produz um alto grau de rejeição por parte dos usuários.

Palavras-chave: usabilidade, checklist, avaliação de usabilidade.

ABSTRACT

The usability aims to support the process of interaction with the user in an effective and efficient way between the user and the interface of a computational system. It can be composed by multiple components being traditionally associated with attributes such as: learnability; efficiency in use; capacity that the system promotes the rememberability of the necessary actions so that the user reaches his or her objectives; control of management of errors minimizing and refuting situations that might occur; and mainly promoting the satisfaction of the user whenever he or she uses it. With the growth of technology, usability started to be a concerning issue, since, due to a market which highly depends on technology, we need products that are easy to both learn and use. The present project aims at the study of the usability for interfaces and the development of an automatized tool of evaluation that offers support to the usability in designing of projects, for developers who need support in the evaluation of interfaces. The evaluation of the usability of a software product is not an absolute guarantee of the product quality, but it certainly is a factor that in its absence produces a high degree of rejection from the users.

Keywords: usability, checklist, usability evaluation.

LISTA DE FIGURAS

Figura 1: Proposta de metodologia de solução.....	16
Figura 2: Atores do Sistema	51
Figura 3: Caso de uso geral	53
Figura 4: Digrama de Robustez (Gerência de Projeto).....	58
Figura 5: Digrama de Robustez (Gerência de Categorias).....	59
Figura 6: Digrama de Robustez (Gerência de recomendações)	60
Figura 7: Digrama de Robustez (Gerência de testes)	61
Figura 8: Diagrama de Atividades – Gerência de Projeto	63
Figura 9: Diagrama de Atividades – Gerência de Categorias	64
Figura 10: Diagrama de Atividades – Gerência de Recomendações.....	65
Figura 11: Diagrama de Atividades – Gerência de Testes	66
Figura 12: Diagrama de Classes de Interfaces.....	68
Figura 13: Digrama de Classes Persistentes	69
Figura 14: Digrama de Classes Persistentes	70
Figura 15: Tela do sistema Ergolist.....	73
Figura 16: Tela do do sistema Ergolist.....	74
Figura 17: Tecnologias utilizadas.....	78
Figura 18: Página inicial.....	80
Figura 19: Cadastro de categorias	81
Figura 20: Cadastro de recomendações	82
Figura 21: Cadastro de projetos.....	83
Figura 22: Associação de recomendações	84
Figura 23: Alteração e exclusão de categorias	85
Figura 24: Alteração de categoria.....	86
Figura 25: Alteração e exclusão de recomendações	87
Figura 26: Alteração de recomendação	88
Figura 27: Alteração e exclusão de projetos.....	89
Figura 28: Alteração de projeto	89
Figura 29: Avaliação	90
Figura 30: Relatório de avaliação.....	92
Figura 31: Página inicial, Módulo Avaliação.....	94
Figura 32: Validação do sistema	95
Figura 33: Validação do sistema 2	95

LISTA DE QUADROS

Quadro 1: Descrição de componentes	16
Quadro 2: Atores do sistema	51
Quadro 3: Casos de uso do sistema	53
Quadro 4: Gerência de Projetos.....	54
Quadro 5: Gerência de Categorias.....	55
Quadro 6: Gerência de recomendações	56
Quadro 7: Gerência de Teste	57
Quadro 8: Tecnologias utilizadas	78
Quadro 9: Grau de impacto	85
Quadro 10: Critério para avaliação.....	90

LISTA DE SIGLAS

ABNT (Associação Brasileira de Normas Técnicas)

ANSI (*American National Standards Institute*)

CSS (*Cascading Style Sheets*)

DHTML: (união das tecnologias HTML e Javascript)

HTML: (*Hypertext Markup Language*)

HTTP: (*HyperText Transfer Protocol*)

IEEE (*Institute of Electrical and Electronics Engineers*)

ISO (*International Organization for Standardization*)

PDF (*Portable Document Format*)

PHP (*Hypertext Preprocessor*)

SGBD: (Sistema de Gerência de Banco de Dados)

SQL: (*Structured Query Language*)

URL: (*Uniform Resource Locator*)

UML: (*Unified Modeling Language*)

XHTML: (*eXtensible Hypertext Markup Language*)

W3C (*The World Wide Web Consortium*)

SUMÁRIO

1. INTRODUÇÃO.....	12
1.1 O Problema	13
1.2 Justificativa	14
1.3 Objetivos.....	14
1.3.1 Geral	14
1.3.2 Específicos.....	15
1.4 Arquitetura da Solução.....	15
1.4.1 Descrição dos Componentes da Arquitetura da Solução do Problema	16
1.5 Delimitação.....	17
1.6 Metodologia Científica	17
1.7 Estrutura da Monografia.....	18
2. USABILIDADE.....	20
2.1 Técnicas de avaliação	24
2.1.1 Preditivas ou analíticas	25
2.1.2 Objetivas ou empíricas	27
2.1.3 Prospectivas	28
2.1.4 Métodos de inspeção	28
2.1.5 Métodos de teste com usuários.....	32
2.1.6 A escolha da técnica mais adequada	34
2.2 Checklist.....	37
2.3 A formação do conteúdo de um <i>checklist</i>	38
2.3.1 As Recomendações do W3C	38
2.3.2 Heurística 1: Visibilidade e reconhecimento do estado ou contexto atual, e condução do usuário.	39
2.3.3 Heurística 2 - Projeto estético e minimalista.	40
2.3.4 Heurística 3 - Controle do usuário.....	41
2.3.5 Heurística 4 - Flexibilidade e eficiência de uso.....	42
2.3.6 Heurística 5 - Prevenção de erros	43
2.3.7 Heurística 6 – Consistência	44
2.3.8 Heurística 7 - Compatibilidade com o contexto	45
2.4 Ergolist.....	46
2.5 Conclusão	48
3. UML.....	50
3.1 Atores.....	50
3.2 Casos de Uso.....	52
3.2.1 CSU001 Gerência de Projetos	54
3.2.2 CSU002 Gerência de Categorias	54
3.2.3 CSU003 Gerência de recomendações.....	55

3.2.4 CSU004 Gerência de Teste	56
3.2.1.1 Diagrama de classes de análise (Gerência de Projeto)	57
3.2.2.1 Diagrama de classes de análise (Gerência de Categoria)	58
3.2.3.1 Diagrama de classes de análise (Gerência de Recomendações).....	59
3.2.4.1 Diagrama de classes de análise (Gerência de teste)	60
3.3 Diagrama de atividades.....	61
3.3.1 Gerência de Projeto	62
3.3.2 Gerência de Categorias.....	63
3.3.3 Gerência de Recomendações	65
3.3.4 Gerência de testes	66
3.4 Diagrama de Classes de Interfaces.....	67
3.5 Diagrama de Classes Persistentes	68
3.6 Conclusão	70
4 DESENVOLVIMENTO E VALIDAÇÃO	72
4.1 A Coleta de Informações.....	72
4.2 Análise de softwares existentes.....	73
4.3 Histórico do Desenvolvimento	74
4.4 Apresentação do sistema.....	79
4.4.1 Módulo Gerência	79
4.4.2 Módulo Avaliação	89
4.5 Validação	92
4.5.1 Validação de usabilidade.....	92
4.5.2 Validação das funcionalidades	93
4.6 Conclusão	96
5 CONCLUSÃO.....	97
REFERÊNCIAS	99

1. INTRODUÇÃO

Não é nenhuma novidade dizer que a internet revolucionou o mundo, quando permitiu a comunicação entre vários usuários além de se tornar a maior rede de informações da atualidade.

Com o crescimento da internet começaram a ser oferecidos os mais variados serviços relacionados ao novo mercado, como *home page* pessoal, *sites* de comércio eletrônico, *sites* de busca, de entretenimento, institucionais entre muitos outros.

O número de domínios, que é o nome de uma área reservada num servidor Internet que corresponde ao endereço numérico de um *website*, cadastrados no mundo chega a impressionante cifra de 82,9 milhões segundo uma pesquisa feita pelo site domains.info (2005).

Atrelado a este crescimento começam a surgir os problemas na utilização desta tecnologia. Antes do advento da internet milhares de usuários já padeciam pelo projeto comprometido das interfaces de seus sistemas, com o uso da rede de informações pela internet este problema piorou e tornou-se globalizado.

A usabilidade busca ajudar o processo de interação com o usuário, pois visa transmitir de forma eficaz e eficiente a interação do usuário com a interface e começa a ser fonte de preocupação, uma vez que com um mercado tão dependente da tecnologia precisamos de produtos que sejam fáceis de aprender e utilizar.

Nielsen (1998) definiu usabilidade como uma medida da qualidade da experiência do usuário ao interagir com alguma coisa - seja um site na Internet, um aplicativo de software tradicional, ou outro dispositivo que o usuário possa operar de alguma forma.

Apesar de a usabilidade ter surgido no início da década de 80 poucos são os esforços que permitem disseminar este conhecimento. Conjuntos de recomendações, normas e *guidelines* dificilmente chegam ao conhecimento do projetista de interfaces.

1.1 O Problema

Com a dependência de tecnologia do mercado, os produtos desenvolvidos são cada vez mais complexos e difíceis. O usuário encontra uma grande dificuldade de interação com a interface tendo dificuldade em resolver problemas que de certa forma seriam simples se a interface fosse bem projetada.

Há muito tempo são publicadas recomendações sobre a usabilidade da interface, mas aplicar estas recomendações exige do avaliador o conhecimento do especialista em usabilidade. O processo para fazer uma avaliação é caro e depende da competência no assunto do avaliador. As recomendações na forma de listas ou mesmo *checklist* são normalmente genéricas, em outras palavras o *checklist* que é oferecido poderia ser aplicado em um software educacional ou mesmo em um produto desenvolvido para automação comercial. Mas cada segmento de software possui características próprias e que não serão provavelmente abordadas durante a avaliação tendo em vista a generalidade da proposta. Nestes casos a avaliação, mesmo sua eficiência, é comprometida, pois os grandes problemas podem residir exatamente nos detalhes do produto relacionados ao escopo do software.

Outro problema é que a maioria dos *checklist's* disponíveis é manual, não objetivando seu relatório de aplicação, ficando isto a critério do avaliador que pode ou não ter experiência em sua aplicação.

1.2 Justificativa

Este projeto se justifica pela possibilidade de oferecer aos desenvolvedores uma ferramenta que irá permitir uma avaliação eficiente dos requisitos de usabilidade de interface, de forma a melhorar a interação do usuário final.

Por outro lado será possível a um usuário não especialista aplicar o *checklist* com grandes chances de sucesso. O uso de recomendações voltadas ao tipo de produto deve gerar um maior grau de alcance nos problemas apresentados, uma vez que poderá ser específico para o tipo de produto.

Outra questão importante é o custo desta avaliação. O mesmo deve ser acessível, uma vez que o desenvolvedor mesmo não tendo o conhecimento sobre o assunto, estará fazendo uso de uma base de recomendações compiladas para o produto em questão.

Um ponto a salientar é que a avaliação na forma de um relatório, servirá como resultado desta etapa de testes e balizamento para futuras melhorias no projeto.

1.3 Objetivos

A seguir são descritos os objetivos do projeto.

1.3.1 Geral

Desenvolvimento de um *checklist* automatizado para avaliação de usabilidade.

1.3.2 Específicos

Os objetivos específicos deste trabalho são apresentados abaixo:

1. Prover o aprofundamento nos estudos na área de usabilidade;
2. Apoiar a avaliação de interfaces com foco na usabilidade;
3. Possibilitar que um usuário leigo em usabilidade realize uma avaliação de usabilidade por meio de uma ferramenta *checklist*;
4. Diminuir o custo da avaliação por meio do *checklis*;
5. Possibilitar o uso de um *checklist* especializado para cada tipo de produto.

1.4 Arquitetura da Solução

A proposta de solução apresenta uma ferramenta de apoio a avaliação em usabilidade, viabilizada a partir de estudos aprofundados e que corresponda às necessidades existentes no mercado na forma de um *checklist*.

As figuras abaixo representam à solução arquitetural do problema.

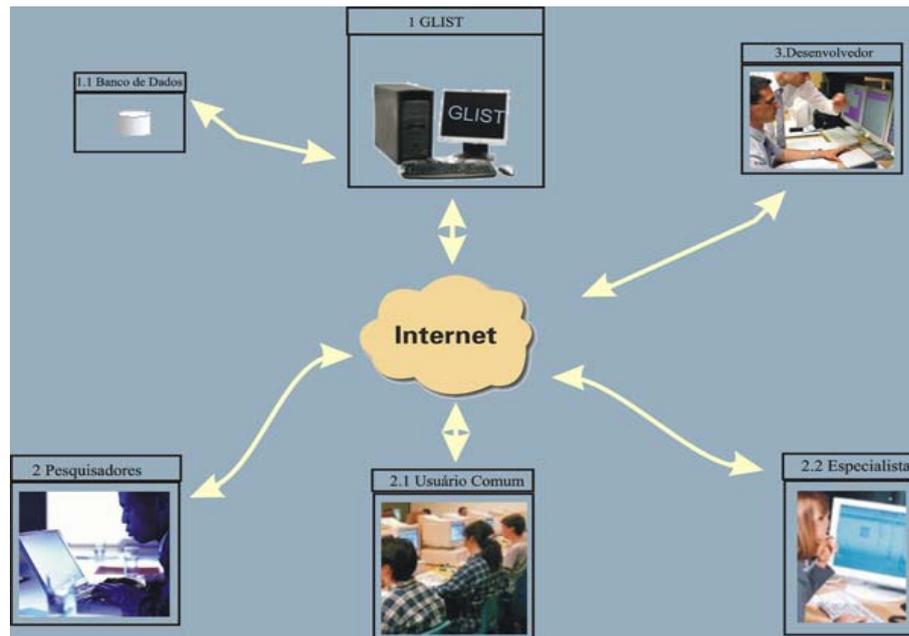


Figura 1: Proposta de metodologia de solução
Fonte: autor do projeto

A arquitetura apresentada mostra de forma simplificada o processo de funcionamento e utilização do GLIST.

O GLIST está disponibilizado na *Web*. Pesquisadores e especialistas na área (com permissão para tal) podem fazer a inserção de dados no sistema formando a base de recomendações que é armazenada num banco de dados. Desenvolvedores podem acessar o sistema através da *Web* e por meio do *checklist* realizar uma avaliação eficiente e segura. O GLIST deve estar disponível para qualquer usuário inclusive não especialista proporcionando-lhes a possibilidade de avaliar seu produto.

1.4.1 Descrição dos Componentes da Arquitetura da Solução do Problema

A seguir serão apresentados os componentes da arquitetura:

Quadro 1: Descrição de componentes
Fonte: Autor do projeto

Componente	Detalhamento
Base de Dados	Responsável pela manutenção e armazenamento dos dados do sistema

GLIST	<i>Checklist</i> automatizado
Pesquisador	Responsável pela alimentação de dados na base do sistema
Usuário Comum	Usuário final do <i>checklist</i> sendo sua habilidade exclusivamente relacionada às tarefas do seu trabalho na empresa.
Especialista	Usuário que utilizara o GLIST para avaliar as suas interfaces.
Desenvolvedor	Programador ou Analista de sistemas sem conhecimento de usabilidade.

1.5 Delimitação

Este trabalho apresenta as seguintes delimitações decorrentes:

- O desenvolvimento da ferramenta deve ser sobre uma arquitetura *open source*;
- A base de dados será populada a partir de outros projetos, sendo que este projeto não prevê a pesquisa de recomendações para popular à base de dados;
- A ferramenta será desenvolvida para ser utilizada a partir da internet sendo disponibilizada ao público em geral.

1.6 Metodologia Científica

O delineamento da pesquisa quanto ao tipo, segundo a Universidade do Sul de Santa Catarina (2003, p. 41) “[...] consiste em informar qual o desenho que a pesquisa terá, ou seja, se a pesquisa será bibliográfica, ou descritiva, ou experimental, ou estudo de caso, ou documental, etc.”.

O estudo desenvolvido neste projeto é considerado, do ponto de vista da sua natureza, uma aplicação prática a fim de conduzir à solução de um problema específico. Pela forma de abordagem do problema classifica-se como uma pesquisa qualitativa, que consiste pelo pesquisador sendo o instrumento chave da pesquisa. Ressaltando os aspectos relacionados aos objetivos, é considerada exploratório, tendo em vista a utilização de estudos

de caso e pesquisa bibliográfica. Pelos procedimentos técnicos classifica-se como uma pesquisa bibliográfica.

1.7 Estrutura da Monografia

Este projeto está subdividido em 5 capítulos. A seguir, será apresentada a estrutura deste trabalho:

Capítulo 1 - Introdução

Este capítulo apresenta a introdução, justificativa, objetivos, problemas e a solução proposta como trabalho.

Capítulo 2 – Usabilidade

Foi desenvolvido neste capítulo o domínio do problema relacionado à usabilidade e a testes em usabilidade.

Capítulo 3 – Modelagem

Estão descritos neste capítulo os passos realizados durante a modelagem, apresentando gráficos e descrições necessárias para a compreensão do sistema.

Capítulo 4 – Desenvolvimento e Validação

Contem a descrição dos passos que tornaram possível a implementação do protótipo bem como os resultados de sua validação.

Capítulo 5 – Conclusão

Neste capítulo estão apresentadas as conclusões e considerações sobre a monografia.

2. USABILIDADE

A usabilidade é definida como “*a capacidade que um sistema interativo oferece a seu usuário, em um determinado contexto de operação, para a realização de tarefas, de maneira eficaz, eficiente e agradável*” (ISO 9241, 1993).

Não se pode falar em usabilidade sem pensar na forma como as pessoas se comunicam e o impacto que um sistema tem para estes indivíduos.

Um sistema que ajuda o usuário na condução de sua tarefa faz com que o indivíduo realize o seu trabalho com mais eficiência, produtividade e satisfação. É importante que o usuário tenha satisfação ao trabalhar com o sistema, assim às pessoas se sentirão bem em utilizá-lo.

Constantine e Lockwood (et all, 1999) define cinco aspectos fundamentais para a construção da usabilidade:

- Facilidade na aprendizagem;
- Facilidade de memorização;
- Eficiência no uso;
- Confiabilidade no uso;
- Satisfação do usuário;

Tais características podem ser vistas como cinco faces da usabilidade, onde se tem aspectos diferentes do sistema e da sua relação de usuário que juntos contribuem para a promoção da usabilidade.

No entanto Nielsen (1999) disponibilizou alguns princípios genéricos de usabilidade que são regras gerais para descrever propriedades comuns de interfaces utilizáveis. São eles:

- Visibilidade do estado do sistema – o sistema deve manter o usuário sempre informado sobre o que está acontecendo, por meio de *feedbacks* apropriados dentro de um tempo razoável.
- Compatibilidade entre o Sistema e o Mundo real – o sistema deve falar o idioma do usuário, com palavras, frases e conceitos familiares ao usuário. Deve fazer com que a informação apareça em uma ordem lógica e natural.
- Controle e liberdade do usuário – freqüentemente o usuário escolhe funções ou seleciona opções por engano, neste caso é necessário oferecer uma “saída de emergência”, para deixar o estado não desejado sem ter que passar por um diálogo muito longo. Opções que permitam desfazer ou fazer novamente, cancelar, limpar tornam o diálogo menos traumático.
- Consistência e padrão – o usuário não deve encontrar palavras, situações ou ações diferentes que significam a mesma coisa.
- Prevenção de erros – boas mensagens e um projeto bem avaliado impedem que um problema aconteça.
- Reconhecimento – torna os objetos, ações e opções visíveis. O usuário não deve lembrar-se da mesma informação de uma parte do diálogo para outro, As instruções devem ser visíveis e recuperáveis.

- Flexibilidade – permite o uso de aceleradores de ações. O botão de busca pode minimizar o tempo de busca de um produto em um site.
- Projeto estético – informações irrelevantes ao conteúdo do site devem ser evitadas. Informações desnecessárias tiram a atenção do usuário de informações pertinentes.
- Ajuda ao reconhecimento, diagnóstico e recuperação de erros – mensagens de erro devem ser expressas em linguagem clara, indicando o problema e propondo uma solução.
- Ajuda e documentação – oferecer ajuda e documentação é fundamental em sites complexos. Todas as informações devem ser de fácil acesso e sua apresentação focada na tarefa de forma concreta e sucinta.

Cybis (2005) define que a usabilidade depende do contexto em que um sistema é operado.

“Assim, um sistema pode proporcionar boa usabilidade para um usuário experiente, mas péssima para novatos, ou vice e versa. Pode ser fácil de operar uma vez lá que outra, mas difícil, se for utilizado no dia a dia. Pode dar prazer se acessado por conexões rápidas, mas causar ansiedade insuportável se acessado de casa. Martin Maguire, em seu artigo publicado em 2001, fala das inter-relações entre contexto de operação e usabilidade e da necessidade de especificar o contexto de uso para o qual uma interface está sendo concebida, e no qual ela será testada. Mas como conceber algo amigável em tantas situações diferentes? A adaptabilidade é uma das boas qualidades de uma interface com o usuário. Isto permitirá que diferentes usuários, em diferentes estágios de competência, em diferentes tarefas e em diferentes ambientes físicos tecnológicos e organizacionais, possam alcançar seus objetivos com eficácia, eficiência e satisfação.”

Hix (1993) defende que o desenvolvimento de um sistema interativo deve contar com três grupos de profissionais integrados:

- a) especialistas no domínio do problema: pessoas que possuem um profundo conhecimento da área que a aplicação interativa pretende suportar;

- b) projetistas de software de interface: profissionais da área de informática, projetistas de software, engenheiros de software e programadores;
- c) projetistas de interação com usuário: usuários, projetistas de interação, avaliadores, e especialistas em fatores humanos e em documentação.

Browne (1992) chama a atenção para o fato de que, atualmente, os sistemas têm sido projetados, desenvolvidos e testados principalmente em relação as suas funcionalidades: tal filosofia de projeto garante que o software atenda aos requisitos funcionais, mas não garante que seja utilizável pelos usuários pretendidos.

Uma interface com as seguintes características tem grandes possibilidades de possuir problemas de usabilidade afirma Hix (1993), as características são: a interface ser projetada por pessoal de software e não por especialista em interface homem-computador, ser desenvolvida por decomposição funcional estritamente *top-down*, não ser desenvolvida para atender especificações de usabilidade documentadas e mensuráveis, não ser desenvolvida como foi prototipada, não ser desenvolvida através de um processo de refinamentos iterativos e não ser avaliada empiricamente.

Os problemas de usabilidade das interfaces são conseqüências diretas de algumas características dos projetistas. Tais como, carecer de conhecimento as tarefas de usuário, carecer uma metodologia de concepção para interface homem computador, conhecer o software segundo orientação funcional e não operacional, não prever erros humanos, não avaliar com precisão as transações de diálogo combinatório, homogeneidade na concepção, considerar o computador como um fim em si mesmo. (SCAPIN, 1986)

Fisher (1989) afirma que “*o software de hoje é tão complexo que técnicas melhores de comunicação são uma necessidade, e não um luxo*”.

Segundo Hix (1993) a interação homem-computador deve constituir-se como uma parte integrante da engenharia de software. Em concordância com esta ideia, De Waal (1990) diz que é de grande importância que o conhecimento e o saber desenvolvidos por ergonomistas e psicólogos cognitivos sejam embutidos no processo de projetar interfaces.

2.1 Técnicas de avaliação

Segundo Dix (1993) a avaliação de usabilidade tem três objetivos principais: avaliar a extensão das funcionalidades do sistema, avaliar os efeitos da interface dos usuários (facilidade de aprendizagem, facilidade e eficiência de uso e efetivo suporte a tarefa) e identificar problemas específicos de usabilidade.

Antes de começar uma avaliação de usabilidade, devemos saber algo sobre a interface do software, para poder analisar a relação entre software e usuário. Não é necessário saber tudo, mas é preciso saber a resposta a poucas perguntas básicas:

- O que os usuários pretendem realizar?
- Como o sistema irá suprir as necessidades do usuário?

Para desenvolver um sistema que evolui através de novas atualizações ou novas versões, é interessante saber a resposta de algumas perguntas adicionais.

- O que o sistema não está realizando e é ineficaz?
- Como o sistema poderia ser feito para ser mais eficiente no suporte utilizado?

As respostas a essas perguntas podem ser procuradas de várias maneiras, principalmente com o cliente. (CONSTANTINE e LOCKWOOD et al, 1999).

Para Nilsen (1993) um método de avaliação visa encontrar problemas de usabilidade. Um problema de usabilidade é um aspecto do sistema e/ou da demanda sobre o

usuário, que torna o sistema ineficiente, difícil de usar ou impossível de ser operado pelo usuário.

Segundo Luzzardi (2003) a avaliação permite:

“(a) constatar, observar e registrar problemas efetivos de usabilidade durante a interação; (b) calcular métricas objetivas para eficácia, eficiência e produtividade do usuário na interação com o sistema; (c) diagnosticar as características do projeto que provavelmente atrapalham a interação por estarem em desconformidade com padrões implícitos e explícitos de usabilidade; (d) prever dificuldades de aprendizado na operação do sistema; (e) prever os tempos de execução de tarefas informatizadas; (f) conhecer a opinião do usuário em relação ao sistema e (g) sugerir as ações de re-projeto mais evidentes face aos problemas de interação efetivos ou diagnosticados.

Durante a última década, a utilização de um grande número de métodos de avaliação de usabilidade tornou o planejamento de um projeto mais curto e mais barato. Geralmente, o esforço e o custo envolvido no teste com usuários reais tem sido visto pela comunidade de desenvolvimento como muito proveitoso no desenvolvimento de um novo *software*. “

Rubin (1994) distingue três tipos de técnicas para avaliação de interfaces gráficas.

São elas preditivas, objetivas e prospectivas.

2.1.1 Preditivas ou analíticas

Rubin (1994) afirma que as técnicas preditivas buscam prever os erros de projeto de interfaces sem a participação direta de usuários, ou seja, baseada no conhecimento de um especialista. As técnicas deste tipo são:

- Inspeção cognitiva - nesse método de avaliação o analista desenvolve uma tarefa específica seguindo a lógica que um usuário teria para desenvolvê-la. Ele, então, deverá responder a um conjunto de questões para cada ação a fim de observar se os princípios foram aplicados. Essas ações e o *feedback* são comparados com os objetivos e conhecimentos do usuário e as discrepâncias entre as expectativas do mesmo e os passos seguidos pela interface são

percebidas, não apenas identificando os problemas, mas também sugerindo as razões para os mesmos. A inspeção cognitiva é um método baseado na teoria cognitiva em que é analisado o processo mental do usuário. Esse método que pode ser realizado durante a construção do *software*, não requer especialistas em ciência cognitiva ou experiente *designer* de interfaces, requer poucos recursos, demanda pouco tempo e esforço de um protótipo do *software*.

- Avaliação heurística - é um método de inspeção sistemático, cujo objetivo é identificar problemas de usabilidade que, posteriormente, serão analisados e corrigidos ao longo do processo de desenvolvimento do sistema. É realizada através de aproximações progressivas onde cada estágio do caminho é percorrido e avaliado e, então, especula-se sobre a natureza dos caminhos a seguir para se aproximar do objetivo de encontrar o maior número possível de problemas de usabilidade. Esse método pode ser usado em qualquer estágio do ciclo de desenvolvimento de um sistema interativo. Esse tipo de avaliação só pode ser realizada por um especialista em interface. Depende da experiência do especialista para fazer a avaliação.
- Inspeção de conformidade - baseada na confrontação com princípios, *guidelines*, recomendações e normas (aplicação de *checklists*). O *checklist* é uma ferramenta para avaliação ergonômica de um *software* a qual verifica a conformidade da interface de um sistema interativo com recomendações ergonômicas provenientes de pesquisas aplicadas. O *checklist* trata de aspectos gerais de uma avaliação ou também oportuniza a focalização de uma lista de questões específicas e detalhadas que conduzem o avaliador durante o processo de avaliação. Pode-se desenvolver versões personalizadas ou especializadas de um *checklist* a partir de recomendações genéricas. O LabiUtil, Laboratório de

Utilizabilidade da Universidade Federal de Santa Catarina, administra um serviço na *Web* chamado *ErgoList* que se constitui de uma base de conhecimento em ergonomia associada a um *checklist* para inspeção ergonômica de interfaces homem-computador.

2.1.2 Objetivas ou empíricas

Rubin (1994) afirma que as técnicas objetivas buscam constatar os problemas a partir da observação do usuário interagindo com o sistema. Tais técnicas podem ser de dois tipos:

- Sistemas de monitoramento - os sistemas de monitoramento visam uma observação direta com os usuários. Estes sistemas são utilitários e permanecem residentes na máquina do usuário, juntamente ao aplicativo em teste (*MS Camcorder* ou *Lotus Screen Can*). Sua finalidade é registrar o usuário em seu momento real de trabalho, assim observando todos os aspectos de interação do usuário com o sistema. Existem pontos positivos neste sistema, pois mesmo que os usuários saibam que estão participando, eles não ficam inibidos e em contrapartida, não há a possibilidade de incentivar ou gravar verbalizações dos usuários. Esses sistemas são limitados tecnicamente em relação às ferramentas de espionagem, pelo fato da diversidade de ambientes de programação. Para se evitar dados em excesso, sugere-se que o tempo do teste seja bem planejado pelos avaliadores.
- Ensaios de interação - os ensaios de interação são testes realizados com os usuários do sistema. É uma simulação de uso do mesmo em que são apresentadas algumas tarefas para o usuário realizá-las. Os mesmos são

acompanhados pelos avaliadores que analisarão os comandos dados, os erros cometidos, e o comportamento do usuário. Os ensaios de interação identificam problemas de interação de mais alto nível, dificilmente identificados por outros métodos.

2.1.3 Prospectivas

Rubin (1994) afirma que as técnicas prospectivas buscam a opinião do usuário sobre a interação com o sistema através da aplicação de questionários.

Para avaliar a usabilidade muitas vezes faz-se necessário saber o seu real contexto de uso. Para isso, deve ser feito o levantamento de informações a respeito do usuário desta interface, as tarefas a serem realizadas e o ambiente onde ocorre a interação entre usuário e sistema. Para os instrumentos de coleta de dados são usados questionários ou entrevistas.

Depois de terem sido identificados os usuários típicos, as tarefas típicas e os ambientes organizacionais, físicos e tecnológicos. O contexto de avaliação deve ser compatível com esse contexto de uso. O próximo passo é identificar os métodos de avaliação de usabilidade mais adequados para o contexto analisado. O conjunto variado de métodos existentes na literatura pode ser subdividido em dois grandes grupos: métodos de inspeção e métodos de teste com usuários.

2.1.4 Métodos de inspeção

A classificação oferecida por Ruben não é uma unanimidade no meio científico

outros autores como Silva (2004) sugere um outro modelo de distribuição das técnicas de avaliação.

Segundo Silva (2004) os métodos de inspeção, ou métodos analíticos ou prognóstico, caracterizam-se pela não participação direta dos usuários do sistema na avaliação. Os avaliadores que adotam esses métodos são especialistas em usabilidade ou projetistas de sistemas, e baseiam-se em regras, recomendações, princípios e conceitos previamente estabelecidos para identificar os problemas de usabilidade. Entre os principais métodos de inspeção identificados na literatura, estão:

- Inspeção de usabilidade formal - é uma adaptação para a avaliação de usabilidade, da metodologia tradicional de inspeção de software para formalização e registro de problemas. Este método é geralmente adotado na fase preliminar de desenvolvimento do projeto, pois faz a detecção inicial de problemas em usabilidade.
- Inspeção ou percurso pluralístico- a inspeção constitui-se de reuniões entre usuários, projetistas de sistemas e especialistas em usabilidade, onde são analisados os cenários das tarefas e avaliados cada um dos elementos da interação do usuário com o sistema. Os dados coletados representam as opiniões ou preferências dos participantes. Pode ser usada antes que o sistema esteja disponível ou até mesmo antes de um protótipo. Como materiais de apoio à avaliação, são utilizados esquemas, anotações, desenhos, painéis e cartões para representar as diversas telas do sistema. A inspeção pluralística é mais usada nos estágios iniciais de desenvolvimento de um sistema.
- Inspeção de componentes - é analisado apenas um conjunto de componentes, características ou módulos do sistema para realização de uma determinada tarefa. A partir de um cenário preestabelecido, são analisados e identificados os

componentes do sistema que seriam utilizados para a realização da tarefa. Utilizar essa análise necessita verificar a disponibilidade, facilidade de compreensão e utilidade de cada componente. É indicado para os estágios intermediários de desenvolvimento dos sistemas, sendo necessário conhecer as funções do sistema e os componentes utilizados pelos usuários para terminar determinada tarefa. Esse método de inspeção busca saber se tais componentes são facilmente utilizáveis pelos usuários.

- Inspeção de consistência - garante a consistência de um conjunto de sistemas relacionados a uma tarefa ou cenários. A equipe de inspeção se reúne para analisar os pontos fortes e fracos das interfaces de cada sistema, para identificar as melhores opções para serem implantadas em todo o conjunto. Conhecida também como revisão de projeto, é mais utilizada nas fases preliminares de desenvolvimento, para não ter que fazer uma reestruturação completa para atingir o padrão desejado.
- Inspeção ou percurso cognitivo - é a técnica de revisão, onde se constroem cenários de tarefa, a partir de uma especificação ou protótipo, e verificam a interface como se fosse o usuário que estivesse utilizando o sistema pela primeira vez. Esse método tem o enfoque na avaliação da facilidade de aprendizado proporcionada pelo sistema e a identificação dos processos cognitivos que se estabelecem quando o usuário realiza a tarefa pela primeira vez. É indicado para os estágios iniciais de desenvolvimento, porque pode ser adotado mesmo quando existem apenas especificações do sistema a ser avaliado. (SILVA 2004)
- Inspeção baseada em padrões - verifica a conformidade do sistema em relação aos padrões da empresa, sendo adotado por especialista com conhecimento em

cada padrão específico. A inspeção é realizada por meio da confrontação de cada elemento do produto com o padrão. Os padrões verificados na avaliação de um sistema interativo podem pertencer a um conjunto de regras ou recomendações estabelecidas por organismos internacionais, tais como ISO (International Organization for Standardization), IEEE (Institute of Electrical and Electronics Engineers), ABNT (Associação Brasileira de Normas Técnicas), ANSI (American National Standards Institute), W3C (The World Wide Web Consortium), etc. A inspeção é adequada para os estágios intermediários de desenvolvimento, cujo projeto baseou-se, desde o princípio, em um ou mais dos padrões existentes.

- o Inspeção baseada em guias de recomendações e guias de estilos - normalmente é utilizado em conjunto com outros métodos de avaliação, como por exemplo, a avaliação heurística e a inspeção de consistência. Os guias usados como um conjunto de requisitos, critérios ou princípios básicos a serem verificados. Os guias de estilos são publicações com descrições mais detalhadas de elementos interativos específicos de um sistema, tais como menus, janelas e caixas de entrada de dados. O guia de recomendações, por outro lado, é um documento publicado, de caráter genérico e público, com recomendações geradas e validadas a partir de observações empíricas ou da experiência prática de seu autor. É composta por uma série de requisitos, considerados desejáveis e/ou necessários para atingir certo efeito ou objetivo, mais restritos e específicos do que os itens de um guia de recomendações. Por atender ao contexto do sistema, isto é, seus usuários, tarefas e ambiente, as listas de verificação são consideradas mais eficientes na detecção de problemas de usabilidade do que os guias de recomendações genéricos. Os guias apresentam, portanto,

sugestões e/ou recomendações para melhoria da usabilidade de sistemas com base em situações empíricas anteriores, na padronização de produtos ou na experiência do avaliador ou projetista.

Como já descrito anteriormente por Rubin (1994):

- Avaliação heurística - é um método de inspeção sistemático, cujo objetivo é identificar problemas de usabilidade que, posteriormente, serão analisados e corrigidos ao longo do processo de desenvolvimento do sistema.

2.1.5 Métodos de teste com usuários

Métodos de teste com usuários caracterizam-se pela participação direta dos usuários na avaliação do sistema. Os métodos podem ser realizados através de questionários e entrevistas, ou podem ser empíricos, ao adotar técnicas de observação ou monitoramento do uso do sistema em situações reais.

- Entrevistas e questionários - permitem que o avaliador conheça as experiências, opiniões e preferências dos usuários ao utilizarem o sistema. A partir de perguntas formuladas de acordo com o objetivo do teste, o avaliador interage com os usuários diretamente, facilitando a discussão sobre os temas sugeridos pelas perguntas, ou envia um questionário e aguarda suas respostas, sem interagir com os usuários participantes do teste. As entrevistas são consideradas técnicas mais informais, mas são capazes de medir a ansiedade, a satisfação subjetiva e a percepção dos usuários com maior riqueza de detalhes do que os questionários ou outras técnicas objetivas. Os questionários são úteis quando se tem uma grande quantidade de usuários, dispersos

geograficamente ou segmentados por perfil. Podem-se identificar indícios de problemas de uso do sistema por certo tipo de usuários, em um determinado ambiente operacional ou realizando certa tarefa. Tanto as entrevistas quanto os questionários podem ser usados em qualquer fase do desenvolvimento do sistema, dependendo do tipo de perguntas formuladas. Dentre as classes de entrevistas e questionários existentes, na área de usabilidade, destacam-se os grupos focais (identifica percepções, sentimentos, atitudes e idéias dos participantes a respeito de um determinado assunto, e é realizada por meio de discussões); e os questionários específicos para medir a satisfação dos usuários (desenvolvidos a partir de técnicas psicométricas).

- Testes Empíricos de Usabilidade - conhecidos também como ensaios de interação, são originários da psicologia experimental e são capazes de coletarem dados a partir da observação da interação homem-computador. A coleta de dados está tendo mais ênfase, para os pesquisadores em usabilidade, usando técnicas como a verbalização (os usuários são solicitados a verbalizar seus pensamentos, sentimentos e opiniões enquanto utilizam o sistema para realizar uma ou mais tarefas no sistema); a co-descoberta (dois participantes realizam juntos tarefas designadas pelo avaliador e descrevem seus pensamentos, observa-se também a ajuda mútua); e dados coletados por métodos de medida de desempenho (coleta dados de desempenho dos usuários típicos, utilizando o sistema para a realização de tarefas específicas).

Segundo Cybis (1999) a usabilidade de um sistema está sempre associada às características de determinados usuários. Assim, um problema de usabilidade pode se fazer sentir fortemente em determinados contextos de operação e ser menor ou mesmo imperceptível, em outros.

As atividades ligadas a esta perspectiva do ciclo de usabilidade visam identificar, sejam por prospecção, diagnóstico ou observação, os problemas de usabilidade em interfaces humano-computador e contribuir para a sua eliminação. Entre os objetivos a serem atingidos, pelas avaliações pode-se observar e registrar problemas efetivos de usabilidade durante a interação, calcular métricas objetivas sobre a eficácia, eficiência e produtividade do usuário na interação com o sistema, diagnosticar as características do projeto que provavelmente atrapalhem a interação por estarem em desconformidade com padrões implícitos e explícitos de usabilidade, prever dificuldades de aprendizado na operação do sistema, prever os tempos de execução de tarefas informatizadas, conhecer a opinião do usuário em relação ao sistema; sugerir as ações de re-projeto mais evidentes face os problemas de interação efetivos ou diagnosticados (CYBIS, 1999).

2.1.6 A escolha da técnica mais adequada

Na escolha de técnicas de avaliação é necessário observar as qualidades de cada técnica a partir da comparação dos recursos disponíveis e das expectativas de resultados a serem obtidos a partir da avaliação de usabilidade. Quando se observa as diferentes técnicas percebem-se diferenças relacionadas à quantidade de problemas que cada uma delas é capaz de detectar, da sistematização de seus resultados, do quanto cada uma delas oferece de facilidades para sua aplicação e principalmente do poder de convencimento dos resultados perante a equipe de projetistas das necessidades de mudanças nas interfaces humano-computador. Cybis (1999) destaca alguns aspectos a serem observados:

- Efetividade: refere-se à quantidade de problemas sérios (recorrentes, transponíveis e assimiláveis) identificados.
- Abrangência: refere-se à quantidade de problemas reais de todos os tipos identificados. As inspeções por *checklists* e as avaliações heurísticas são as mais abrangentes;
- Eficiência: é a razão da quantidade de problemas sérios (recorrentes, transponíveis e assimiláveis) identificados em face de quantidade de problemas reais identificados de todos os tipos.
- Produtividade: refere-se a razão entre a quantidade de problemas reais de todos os tipos identificados em relação à quantidade de recursos financeiros (U\$) necessários;
- Sistematização: para esta qualidade concorrem dois fatores igualmente importantes: repetitividade e reproduzibilidade. O primeiro refere-se à medida pela quais os resultados produzidos pela técnica se repetem quando o mesmo avaliador examina o mesmo software algum tempo depois da primeira avaliação. O segundo fator se refere à medida pela quais dois avaliadores diferentes examinando um mesmo software produzem os mesmos resultados. As inspeções por *checklists* são as mais sistemáticas.

Apesar do teste de usabilidade resolver grande parte dos problemas de interação com o usuário, Rubin (1994) afirma que o teste não é tudo para usabilidade e sucesso do produto, e isto é importante para compreender suas limitações. Testes não garantem o sucesso ou produzem provas que o produto será usável. Até mesmo o maior rigor conduzindo formalmente o teste não pode dar cem por cento de certeza, assegurando que o produto será utilizável quando liberado. Aqui estão algumas razões apresentadas por (RUBIN, 1994):

- O teste é sempre uma situação artificial. O teste no laboratório, ou até mesmo o teste de campo, ainda representa uma descrição de uma situação artificial de uso e não a situação em si. O ato de conduzir um estudo pode em si afetar o resultados.

- Resultado de teste não prova que o produto funciona. Mesmo que um plano de administração do tipo de teste que adquire resultados estatisticamente significativo, isto ainda não prova que um produto é eficaz. O significado da estatística é simplesmente uma medida de uma probabilidade de que um único resultado não acontece acidentalmente. Isto não é uma garantia, e depende muito de que modo o método foi administrado.
- Os participantes são, raramente, representantes do público alvo. Participantes são somente os representantes capacitados para compreender e classificar seu público alvo. O mercado de pesquisa não é uma ciência infalível, e o real usuário fica no final e é frequentemente difícil identificá-lo e descrevê-lo.
- Testes nem sempre são a melhor técnica para usar. Há muitas técnicas pretendidas para avaliar e melhorar os produtos. Por exemplo, em alguns casos que mais técnicas são eficazes em termos de custo, tempo e exatidão para conduzir uma avaliação técnica de um produto ao invés de testá-lo.

Entretanto, apesar destas limitações, testes de usabilidade, quando conduzidos com cuidado e precisão, para as noções apropriadas, no tempo apropriado do ciclo de vida do produto, e como a parte de uma aproximação total do usuário no centro da criação, é um indicador quase infalível de problemas potenciais e o meio de resolvê-los minimiza o risco consideravelmente de liberar um produto instável ou de difícil aprendizagem.

De acordo com a comparação feita por Silveira (2001) a avaliação por *checklist* derivada de guias de recomendações ou critérios ergonômicos é uma boa alternativa à avaliação no início do processo e, quando for necessário envolver desenvolvedores. Pelas suas características, os guias enriquecem a avaliação, mas não devem ser usados como métodos únicos. Para uma aplicação ser completa, seria necessário usar mais que uma técnica de avaliação, escolhidas em função dos objetivos a serem alcançados.

2.2 Checklist

Cybis (1997), utiliza o termo inspeção ergonômica referindo-se a um tipo de avaliação ergonômica em que o analista realiza vistorias baseadas em *guidelines*. E ainda destaca que os resultados produzidos são uniformes, pois inspetores são conduzidos ao exame da interface por meio de uma análise da lista de questões a responder sobre a ergonomia do projeto.

Heemann (1997) diz que “o *Checklist* é uma ferramenta para a avaliação da qualidade ergonômica de um software, que se caracteriza pela verificação da conformidade da interface de um sistema interativo com recomendações ergonômicas provenientes de pesquisas aplicadas.”.

A avaliação realizada por meio de *Checklists* apresenta as seguintes características, segundo Heemann (1997):

- Possibilidade de ser realizada por projetistas, não exigindo especialistas em interfaces homem-computador, pois o conhecimento ergonômico está contido no *Checklist*;
- Sistematização da avaliação, o que garante resultados mais estáveis, mesmo quando aplicada separadamente por diferentes avaliadores, pois as questões/recomendações constantes no *Checklist* sempre serão efetivamente verificadas;
- Facilidade na identificação de problemas de usabilidade, devido à especificidade das questões do *Checklist*;
- Aumento da eficácia de uma avaliação, devido a uma considerável redução da subjetividade normalmente associada a processos de avaliação;
- Redução de custo da avaliação, pois é um método de rápida aplicação.

Segundo Matias (1995), o *Checklist* é uma ferramenta capaz de dar suporte à avaliação preliminar da interface, pois consegue identificar a maior parte dos problemas detectados por uma análise ergonômica completa, que envolva a utilização de outras técnicas e aumento da eficácia da avaliação.

2.3 A formação do conteúdo de um *checklist*

A técnica de avaliação de um *checklist* ergonômico foi desenvolvida para sistematizar, objetivar e facilitar as avaliações em usabilidade, permitindo que o resultado esteja de acordo com os componentes de interfaces humano-computador. Geralmente a base de conhecimento das questões de um *checklist* é formulada a partir de recomendações publicadas, como a W3C e guias de estilos, como o guia de estilos para serviços de informação em ciência e tecnologia via Web (PARIZOTTO, 1997).

Na Web é possível encontrar um *checklist* baseado nas recomendações W3C.

2.3.1 As Recomendações do W3C

O *World Wide Web Consortium* (W3C), é um consórcio de empresas de tecnologia fundada por Tim Berners Lee em 1994 para levar a web, através do desenvolvimento de protocolos comuns e fóruns abertos que promovem sua evolução e asseguram a sua interoperabilidade. O W3C desenvolve tecnologias, denominadas *Web Standards* (ou Padrões Web) para a criação e a interpretação dos conteúdos para web. Na web é possível encontrar um *checklist* baseado nas recomendações da W3C.

Sites desenvolvidos segundo esses padrões, podem ser acessados e visualizados por qualquer pessoa ou tecnologia independente de hardware ou software utilizados, como

celular, PDA, eletrodomésticos, independentemente da plataforma, de maneira rápida e compatível com os novos padrões e tecnologias que possam surgir com a evolução da internet. Apesar de o W3C não ser muito conhecido no Brasil, padrões seus como *Hyper Text Markup Language* (HTML), *eXtensible Hypertext Markup Language* (XHTML) e *Cascading Style Sheets* (CSS) são muito populares, contudo em, muitos casos, são usados de forma errônea devido ao não conhecimento da especificação.

São apresentadas a seguir as heurísticas que originaram o *checklist* da W3C onde Dias (2001) descreve-las como está listada abaixo:

- o Visibilidade e reconhecimento do estado ou contexto atual, e condução do usuário.
- o Projeto estético e minimalista.
- o Controle do usuário
- o Flexibilidade e eficiência de uso
- o Prevenção de erros
- o Consistência
- o Compatibilidade com o contexto

2.3.2 Heurística 1: Visibilidade e reconhecimento do estado ou contexto atual, e condução do usuário.

Dias (2001) refere-se aos meios disponíveis para informar, orientar e conduzir o usuário durante a interação com o portal corporativo.

Em virtude da forma hipertextual, não-linear de interação e da quantidade de páginas disponíveis na Internet, um dos maiores problemas identificados em testes com usuários é sua desorientação. Para diminuir os efeitos dessa desorientação, o portal deve sempre manter o usuário informado quanto à página em que ele se encontra, como chegou até essa página e quais são suas opções de saída, isto é, onde ele se encontra numa seqüência de

interações ou na execução de uma tarefa. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o A página principal do portal deve ser capaz de responder às seguintes perguntas: “Onde estou?” e “O que este portal faz?”.
- o Apresentar em destaque o nome da página principal em todas as páginas componentes do portal, preferencialmente no canto superior esquerdo. Pode-se usar o termo *Home* ou o logotipo da empresa/departamento/projeto, por exemplo.
- o A navegação entre as páginas do portal deve responder às três perguntas: “Onde estou?”, “Onde estive?” e “Para onde posso ir?”.
- o Apresentar a estrutura ou mapa de navegação do portal, ressaltando a página atual onde o usuário se encontra. Por exemplo, o indicativo “Você está aqui!”, como nos mapas turísticos.

2.3.3 Heurística 2 - Projeto estético e minimalista.

Refere-se às características que possam dificultar ou facilitar a leitura e a compreensão do conteúdo disponível no portal. Destacam-se a legibilidade, a estética e a densidade informacional. (DIAS, 2001).

Um programa legível e esteticamente agradável facilita a leitura da informação nele apresentada, melhorando inclusive o desempenho do usuário na realização da tarefa proposta e influenciando seu nível de satisfação durante a interação com o portal. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o Ocupar de 50 a 80% da página com conteúdo (preferencialmente, 80%).
- o Ocupar no máximo 20% da página com informações sobre a navegação.
- o Evitar *frames*, pois diminuem o espaço disponível para apresentação de conteúdo.
- o Usar hipertexto para dividir as informações em várias páginas ou níveis de detalhamento.
- o Usar pequenos parágrafos, subtítulos e listas.
- o Agrupar os diferentes tipos de informações disponíveis na página, apresentando as mais importantes em primeiro lugar.

2.3.4 Heurística 3 - Controle do usuário

Dias (2001) relaciona o controle que o usuário sempre deve ter sobre o processamento de suas ações pelo portal. Os usuários de qualquer sistema interativo esperam deter controle sobre o sistema, fazendo com que este responda a suas solicitações e expectativas. Ações inesperadas do sistema, infundáveis seqüências de entrada de dados, incapacidade ou dificuldade em obter a informação necessária e incapacidade em produzir os resultados desejados contribuem para o aumento da ansiedade e da insatisfação do usuário. As ações de um sistema devem ser reversíveis, o usuário deve ter a possibilidade de desfazer a última ação. Assim diminui a ansiedade do usuário. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o Possibilitar o retorno à página anterior.

- o Possibilitar aos usuários interromper ou cancelar o processamento ou transação atual.
- o Não desviar para outra página, a não ser que o usuário digite *Enter* ou clique com o mouse.
- o Não abrir janelas adicionais.
- o Apresentar em todas as páginas os níveis anteriores da estrutura de navegação (em forma de *links*) até chegar à página atual (em formato textual, sem link). Dessa forma, o usuário poderá retornar mais facilmente às páginas anteriores.

2.3.5 Heurística 4 - Flexibilidade e eficiência de uso

Dias (2001) sugere a capacidade do sistema em se adaptar ao contexto e às necessidades e preferências do usuário, tornando seu uso mais eficiente.

Em função da diversidade de tipos de usuários de um portal, é necessário que sua interface seja flexível para realizar a mesma tarefa de diferentes maneiras, de acordo com o contexto e com as características de cada tipo de usuário. Devem-se fornecer ao usuário procedimentos e opções diferentes para atingir o mesmo objetivo, da forma que mais lhe convier.

Outros procedimentos podem ser adotados para tornar o uso do portal mais eficiente, tais como a eliminação de páginas ou passos desnecessários em uma seqüência para a realização de uma tarefa e o uso de valores padronizados, sem a necessidade de digitação por parte do usuário. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o Não usar páginas sem conteúdo útil, como por exemplo, páginas apenas com mensagens do tipo "Seja bem-vindo ao portal tal".
- o Na identificação de *links*, utilizar termos que expressem o conteúdo das páginas correspondentes. Não utilizar números ou cores para isso.
- o Oferecer serviço de busca para pesquisa das páginas do portal, incluindo a possibilidade de verificação ortográfica dos termos digitados em sua caixa de entrada de dados.
- o Nos resultados de pesquisa do serviço de busca, apresentar os melhores em primeiro lugar, sendo desnecessário o uso de porcentagens ou graus de acerto.

2.3.6 Heurística 5 - Prevenção de erros

Dias (2001) sugere relacionar-se todos os mecanismos que permitem evitar ou reduzir a ocorrência de erros, assim como corrigir os erros que porventura ocorram.

As interrupções provocadas por erros de processamento têm conseqüências negativas sobre a atividade do usuário, prolongando e perturbando a realização de suas tarefas. Quanto menor a probabilidade de erros, menos interrupções ocorrem e melhor o desempenho do usuário. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o Não usar páginas com a expressão “em construção”. O portal deve apresentar apenas o que já está pronto para ser acessado pelo usuário.
- o Não liberar portal parcialmente pronto.

- o Remover dados/páginas desatualizados, como por exemplo, páginas convidando os usuários para participarem de eventos que já ocorreram.
- o Nos serviços de busca, não usar operadores *booleanos* nas pesquisas simples. É recomendável oferecer a possibilidade de operadores *booleanos* apenas em pesquisas avançadas, para serem usadas pelos usuários mais experientes.

2.3.7 Heurística 6 – Consistência

Dias (2001) refere-se à consistência como a homogeneidade e coerência na escolha de opções durante o projeto da interface do sistema (denominação, localização, formato, cor, linguagem).

Um projeto consistente facilita o reconhecimento, o aprendizado, a localização e, por fim, a utilização de um portal por seus usuários. A padronização de formatos, localizações e sintaxe tornam o portal mais previsível, diminuindo a incidência de erros e as dificuldades de aprendizado e compreensão. Quando possível e conveniente padronizar os elementos do sistema quanto a seu formato cor, localização e denominação, para que o usuário identifique mais facilmente situações e elementos similares e realize suas tarefas com maior rapidez. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o Usar sempre a mesma terminologia e a mesma localização de elementos comuns nas páginas de conteúdo, nas páginas de ajuda ao usuário e nas mensagens de erro.
- o Incluir a caixa de entrada de dados do serviço de busca logo no início de cada página, preferencialmente no canto superior direito.

- o O comportamento do cursor deve ser consistente em todos os campos de entrada de dados, isto é, o cursor deve saltar automaticamente de um campo a outro ou aguardar *Enter* ou *Tab* do usuário.
- o Verificar se os títulos ou cabeçalhos das páginas correspondem exatamente aos termos utilizados nos *links* que apontam para essas páginas.

2.3.8 Heurística 7 - Compatibilidade com o contexto

Dias (2001) refere-se à correlação direta entre o portal e seu contexto de aplicação. As características do portal devem ser compatíveis com as características dos usuários e das tarefas que estes pretendem realizar com o sistema.

O desempenho dos usuários de qualquer sistema interativo melhora quando os procedimentos necessários ao cumprimento da tarefa são compatíveis com as características psicológicas, culturais e técnicas dos usuários; e quando os procedimentos e as tarefas são organizados de acordo com as expectativas e costumes dos usuários. O portal deve “falar a linguagem do usuário, com palavras, conceitos familiares, ao invés de termos técnicos que o usuário não entende”. A seguir são relacionadas algumas recomendações segundo Dias (2001).

Recomendações

- o Planejar a estrutura do portal de acordo com o contexto das tarefas realizadas pelos usuários e não com a estrutura organizacional ou com as novidades tecnológicas. A estrutura deve ser determinada pelas tarefas que os usuários pretendem realizar por meio do portal.

- o Evitar estrutura linear (ordem numérica ou alfabética). As informações devem ser apresentadas seguindo uma ordem lógica relacionada à tarefa a realizar.
- o Verificar erros de grafia, tomando como base a gramática do idioma utilizado e o glossário de termos técnicos de uso corrente na instituição.

2.4 Ergolist

O *checklist* disponibilizado pelo LabUtil¹ chamado de ErgoList, utiliza-se dos critérios de Bastien e Scapin (1993) na composição de suas questões, descritas na tabela abaixo.

Presteza- Engloba os meios utilizados para levar o usuário a realizar determinadas ações, como, por exemplo, entrada de dados. Esse critério engloba também todos os mecanismos ou meios que permitem ao usuário conhecer as alternativas, em termos de ações, conforme o estado ou contexto nos quais ele se encontra. O critério se subdivide em quatro subcritérios:

- o Agrupamento por Localização;
- o Agrupamento/distinção por Formato;
- o Feedback;
- o Legibilidade.

¹ Disponível em <http://www.labiutil.inf.ufsc.br/ergolist/index.html>

Concisão - O critério concisão diz respeito a todos os elementos da interface que têm um papel importante na redução da carga cognitiva e perceptiva do usuário e no aumento da eficiência do diálogo. O critério se subdivide em dois subcritérios:

- o Ações Mínimas;
- o Densidade Informacional.

Ações Explícitas - O critério Ações Explícitas do Usuário refere-se às relações entre o processamento pelo computador e as ações do usuário. Essa relação deve ser explícita, isto é, o computador deve processar somente aquelas ações solicitadas pelo usuário e apenas quando solicitado a fazê-lo. O critério se subdivide em um subcritérios:

- o Controle do Usuário.

Flexibilidade - Trata-se, em outros termos, da capacidade da interface de se adaptar às variadas ações do usuário.

Experiência do Usuário - Experiência do usuário diz respeito aos meios implementados que permitem que o sistema respeite o nível de experiência do usuário.

Proteção contra erros - A proteção contra os erros diz respeito aos mecanismos empregados para detectar e prevenir os erros de entradas de dados, comandos, possíveis ações de conseqüências desastrosas e/ou não recuperáveis. O critério se subdivide em dois subcritérios:

- o Mensagens de erro;
- o Correção de erros.

Consistência - Refere-se à forma na quais as escolhas na concepção da interface (códigos, denominações, formatos, procedimentos, etc.) são conservadas idênticas, em contextos idênticos, e diferentes, em contextos diferentes.

Significado dos Códigos - Diz respeito à adequação entre o objeto ou a informação apresentada ou pedida e sua referência. Códigos e denominações significativas possuem uma forte relação semântica com seu referente. Termos pouco expressivos para o usuário podem ocasionar problemas de condução, podendo levá-lo a selecionar uma opção errada.

Compatibilidade - Refere-se ao acordo que possa existir entre as características do usuário (memória, percepção, hábitos, competências, idade, expectativas, etc.) e as tarefas, de uma parte, e a organização das saídas, das entradas e do diálogo de uma dada aplicação.

2.5 Conclusão

Neste projeto, foi possível verificar conceitos de usabilidade e meios de avaliação, como critérios e recomendação para apoiá-las. Qualquer uma dessas avaliações, com suas peculiaridades, resolve problemas relacionados à usabilidade. Ainda é pouco abordada dentro das empresas de software.

Deve-se ter em mente que a elaboração e realização de testes de usabilidade aplicados a um produto, são bons indicadores para expor problemas em potencial. Isto minimiza o risco de disponibilizar no mercado um produto instável e de difícil aprendizado. A técnica para avaliação do produto deve ser definida por um especialista que está executando o projeto de usabilidade, ele deve definir o melhor método ou conjunto de métodos a ser aplicado.

Autores como Nilsen (2000), Shneiderman (1992), Bastien (1993) e Scapin (1986) formularam recomendações para apoio à avaliação em usabilidade de forma genérica.

Quando utilizadas como base para um *checklist* as recomendações ou guias de estilos oferecidos são transformadas em questões onde o avaliador pode responder objetivamente sim, não e não aplicável, evitando a subjetividade nas respostas.

O uso de *checklists* é um meio barato e eficiente de capturar problemas de usabilidade, a usabilidade é um elemento crítico de sucesso no mercado de informática e a realização dos testes de usabilidade é um elemento imprescindível no projeto.

3. UML

UML (*Unified Modeling Language*) é um padrão de linguagem para elaboração da estrutura de projetos de software, podendo ser empregada para visualização, a especificação, a construção e a documentação de sistemas.

A UML não é uma linguagem visual de programação, mas é capaz de representar tudo o que possa ser melhor expresso em termos gráficos já que a linguagem de programação representa o que é melhor em termos textuais.

Esse tipo de linguagem serve para a documentação da arquitetura do sistema e de todos os seus detalhes, ela também proporciona uma linguagem para expressão de requisitos para realização de testes. Para finalizar, a UML oferece uma linguagem para a modelagem das atividades de planejamento do projeto e de gerenciamento de versões. (BOOCH, 2000).

3.1 Atores

O ator representa um conjunto coerente de papéis que os usuários de casos de uso desempenham quando interagem com esses casos de uso. Um ator representa o papel

do ser humano, um dispositivo de hardware ou outro sistema desempenha com o sistema.

(BOOCH, 2000).

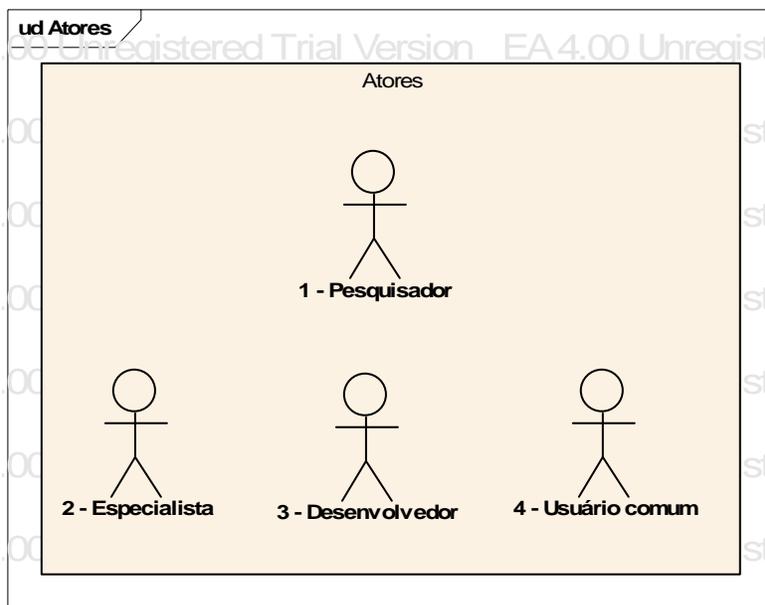


Figura 2: Atores do Sistema

Fonte: autor do projeto

Quadro 2: Atores do sistema

Fonte: Autor do projeto

Nro.	Ator	Descrição
1	Pesquisador	<p>Definição: Indivíduo que realiza a alimentação da base de dados do Glist</p> <p>Frequência de Uso: Diária</p> <p>Objetivo : avaliar a usabilidade de um determinado produto ou popular a base de recomendações.</p> <p>Conhecimento em informática: relativamente alto, porque são pesquisadores na área da computação.</p> <p>Conhecimento no Processo: Domina todo o processo.</p> <p>Grau de escolaridade, Graduado ou em processo de graduação.</p> <p>Permissões de acesso: Terá acesso a todas as funcionalidades do sistema.</p>
2	Especialista	<p>Definição: O especialista será usuário final que poderá fazer avaliação em usabilidade, ou se tiver permissão poderá fazer a alimentação do sistema.</p> <p>Frequência de Uso: Diária</p> <p>Objetivo : avaliar a usabilidade de um determinado produto ou popular a base de recomendações.</p>

		<p>Conhecimento em informática: relativamente alto, são especialista na área de usabilidade.</p> <p>Conhecimento no Processo: Domina todo o processo.</p> <p>Grau de escolaridade, Doutores, Mestres, Graduados.</p> <p>Permissões de acesso: Terá acesso a principio somente a consulta à base de dados, mas com uma permissão poderá ter acesso total do sistema através de uma senha.</p>
3	Desenvolvedor	<p>Definição: Usuário alvo do Glist.</p> <p>Frequência de Uso: Diária</p> <p>Objetivo: melhorar a usabilidade do produto que desenvolve.</p> <p>Conhecimento em informática: relativamente alto, são programadores.</p> <p>Conhecimento no Processo: Baixo.</p> <p>Grau de escolaridade: Graduação, pós-graduado, técnico.</p> <p>Permissões de acesso: Terá acesso some a consulta da base de dados.</p>
4	Usuário comum	<p>Definição: Usuário final.</p> <p>Frequência de Uso: Diária</p> <p>Objetivo: conhecer a usabilidade do produto que utiliza.</p> <p>Conhecimento em informática: médio, porque o usuário que deseja fazer uma avaliação de usabilidade já tem um conhecimento em informática.</p> <p>Conhecimento no Processo: Baixo</p> <p>Grau de escolaridade: Desde fundamental ate pós-graduação</p> <p>Permissões de acesso: Terá acesso somente a consulta.</p>

3.2 Casos de Uso

O caso de uso descreve o que o sistema (ou um subsistema, classe ou interface) faz, mas ele não especifica como isso e feito. Os Diagramas são importantes para visualizar, especificar e documentar o comportamento de um elemento. Isso faz com que os sistemas, subsistemas e classes ficam mais acessíveis e compreensíveis, por apresentarem a visão externa sobre como esses elementos podem ser visualizados no contexto. (BOOCH, 2000).

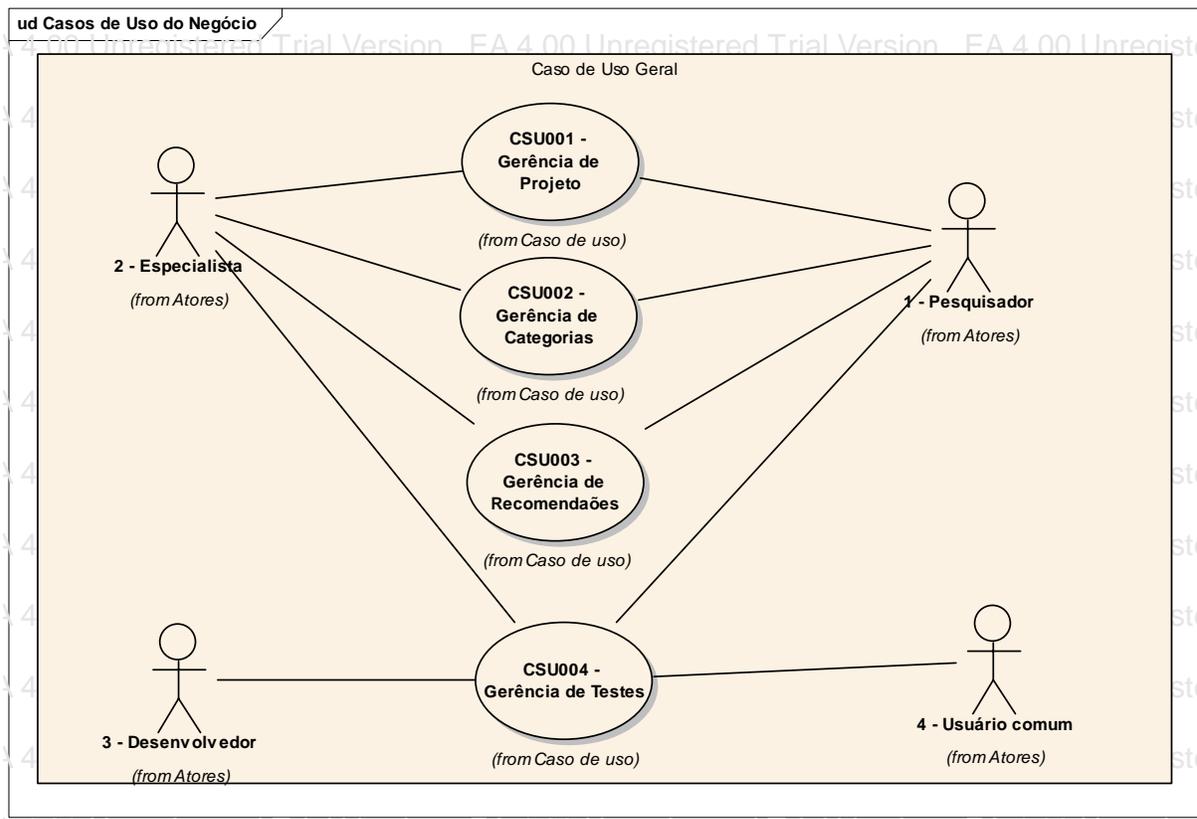


Figura 3: Caso de uso geral
Fonte: Autor do projeto

Quadro 3: Casos de uso do sistema
Fonte: Autor do projeto

Código	Descrição
CSU001	Gerência de Projetos Realiza a inserção, alteração, exclusão de projetos.
CSU002	Gerência de Categorias Realiza a inserção, alteração, exclusão de categorias.
CSU003	Gerência de Recomendações Realiza a inserção, alteração, exclusão de recomendações.
CSU004	Gerência de Testes Realiza com base no projeto escolhido, o teste a ser aplicado.

3.2.1 CSU001 Gerência de Projetos

O caso de uso “CSU001 Gerência de Projetos” tem por objetivo o cadastro de projetos na sua base de dados. Um projeto e a área de atuação que o mesmo estará abrangendo.

Quadro 4: Gerência de Projetos

Fonte: Autor do projeto

Caso de Uso - Gerência de projetos	
<pre> graph LR P((Pesquisador)) --- UC((CSU001- Gerência de Projeto)) E((Especialista)) --- UC </pre>	
Nome	Gerência de Projetos
Identificador	CSU001
Descrição	O pesquisador ou especialista cadastra ou altera um projeto no sistema
Ator Primário	Pesquisado e Especialista
Pré-condição	Conta e senha do usuário devem ser válidas
Fluxo Principal	<ol style="list-style-type: none"> 1. O usuário informa o nome do novo projeto. 2. O sistema permite a associação do projeto com as recomendações 3. O usuário seleciona a opção. 4. O sistema executa a opção desejada.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. O usuário do sistema seleciona um projeto já cadastrado. 2. O sistema exibe a opção para alterar ou excluir o projeto selecionado 3. O usuário cancela as opções acessando o menu.
Pós-condição	Cadastro realizado com sucesso.

3.2.2 CSU002 Gerência de Categorias

O caso de uso “CSU002 Gerência de categorias” tem por objetivo o cadastro de categorias na sua base de dados. As categorias estarão vinculadas aos seus respectivos projetos.

Quadro 5: Gerência de Categorias

Fonte: Autor do projeto

Caso de Uso - Gerência de Categorias	
Nome	Gerência de Categorias
Identificador	CSU002
Descrição	O pesquisador ou especialista cadastra ou altera categorias no sistema
Ator Primário	Pesquisado e Especialista
Pré-condição	Conta e senha do usuário devem ser válidas
Fluxo Principal	<ol style="list-style-type: none"> 1. O usuário informa o nome das categorias. 2. O sistema permite o usuário cadastrar várias categorias. 3. O sistema exibe a opção para finalizar o cadastro. 4. O usuário seleciona a opção. 5. O sistema executa a opção desejada.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. O usuário do sistema seleciona categoria já cadastrada. 2. O sistema exibe a opção para alterar ou excluir a categoria selecionada. 3. O usuário cancela as opções acessando o menu.
Pós-condição	Cadastro realizado com sucesso.

3.2.3 CSU003 Gerência de recomendações

O caso de uso “CSU003 Gerência de recomendações” tem por objetivo o cadastro de recomendações na sua base de dados. As recomendações estarão vinculadas as suas respectivas categorias e projetos.

Quadro 6: Gerência de recomendações

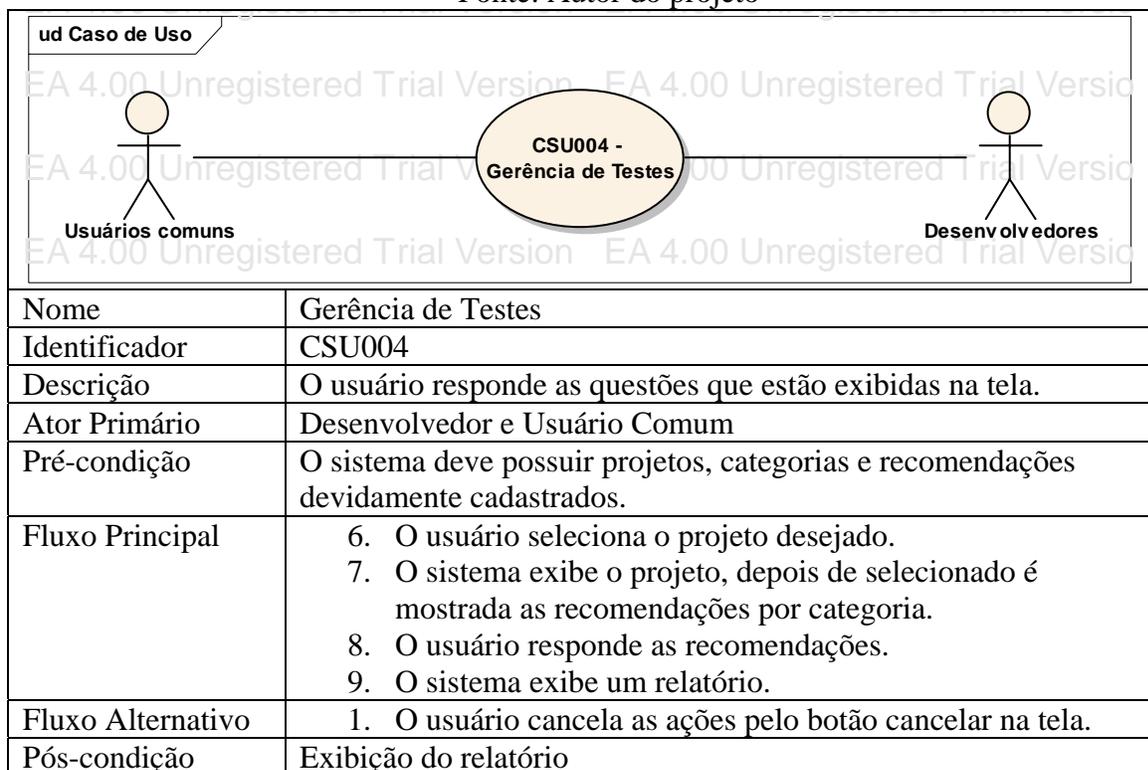
Fonte: Autor do projeto

Ud «subsystem» Gerencia de recomendações	
<pre> graph LR subgraph "Ud «subsystem» Gerencia de recomendações" direction LR P((Pesquisador)) --- UC((CSU003 - Gerência de Recomendações)) E((Especialista)) --- UC end </pre>	
Nome	Gerência de Recomendações
Identificador	CSU003
Descrição	O pesquisador ou especialista cadastra ou altera recomendações no sistema
Ator Primário	Pesquisado e Especialista
Pré-condição	Conta e senha do usuário devem ser válidas.
Fluxo Principal	<ol style="list-style-type: none"> 1. O usuário informa o nome das recomendações. 2. O sistema permite o usuário cadastrar várias recomendações. 3. O sistema exibe a opção para finalizar o cadastro. 4. O usuário seleciona a opção. 5. O sistema executa a opção desejada.
Fluxo Alternativo	<ol style="list-style-type: none"> 1. O usuário do sistema seleciona recomendações já cadastradas. 2. O sistema exibe a opção para alterar ou excluir a recomendação selecionada. 3. O usuário cancela as opções acessando o menu.
Pós-condição	Cadastro realizado com sucesso.

3.2.4 CSU004 Gerência de Teste

O caso de uso “CSU004 Gerência de testes” tem por objetivo montar os testes de acordo com o projeto selecionado pelo usuário, subdivididos por categorias e recomendações.

Quadro 7: Gerência de Teste
Fonte: Autor do projeto



3.2.1.1 Diagrama de classes de análise (Gerência de Projeto)

Representa a estrutura estática da classe Entidade projeto, esta por sua vez, representa os objetos que serão gerenciados pela interface form: Gerência de Projeto.

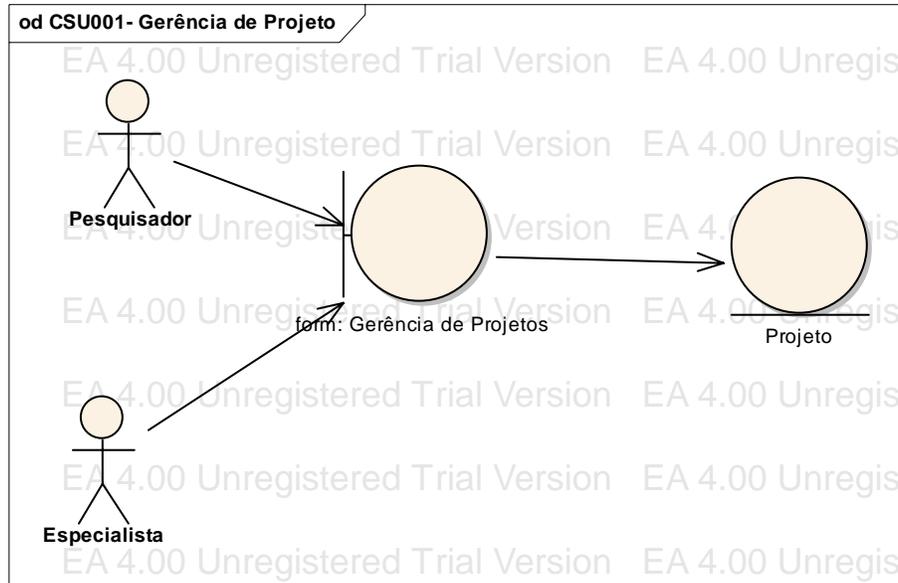


Figura 4: Digrama de Robustez (Gerência de Projeto)

Fonte: Autor do projeto

O sistema apresenta a tela form: Gerência de Projeto, o usuário informa o nome do novo projeto. Os dados são armazenados na classe persistente Projeto.

3.2.2.1 Diagrama de classes de análise (Gerência de Categoria)

Representa a estrutura estática da classe Entidade Categoria, esta por sua vez, representa os objetos que serão gerenciados pela interface form: Gerência de Categoria.

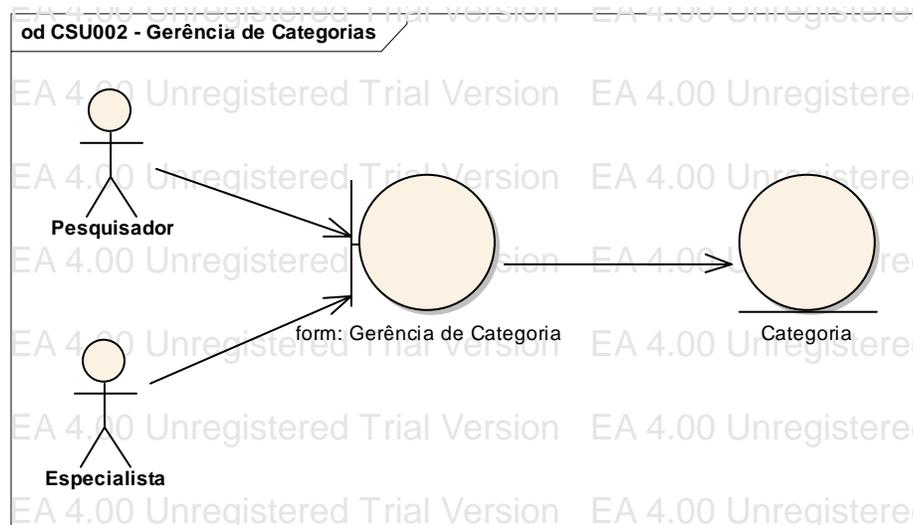


Figura 5: Digrama de Robustez (Gerência de Categorias)

Fonte: Autor do projeto

O sistema apresenta a tela form: Gerência de Categoria, o usuário informa o nome das novas categorias. Os dados são armazenados na classe persistente Categoria.

3.2.3.1 Diagrama de classes de análise (Gerência de Recomendações)

Representa a estrutura estática da classe Entidade recomendações, esta por sua vez, representa os objetos que serão gerenciados pela interface form: Gerência de Recomendações.

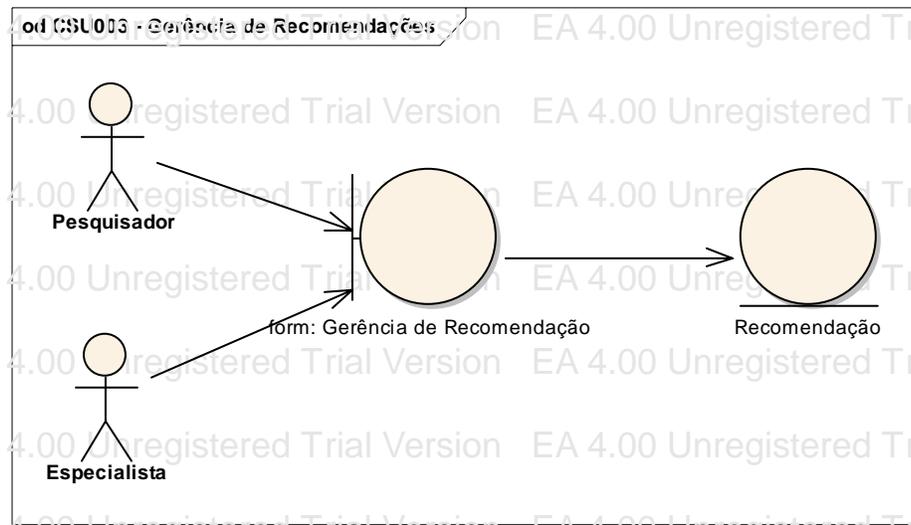


Figura 6: Digrama de Robustez (Gerência de recomendações)
Fonte: Autor do projeto

O sistema apresenta a tela form: Gerência de recomendações, o usuário informa o nome das novas recomendações. Os dados são armazenados na classe persistente Recomendações.

3.2.4.1 Diagrama de classes de análise (Gerência de teste)

Representa a estrutura das classes, esta por sua vez, representa os objetos que serão gerenciados pela interface form: Gerência de Testes.

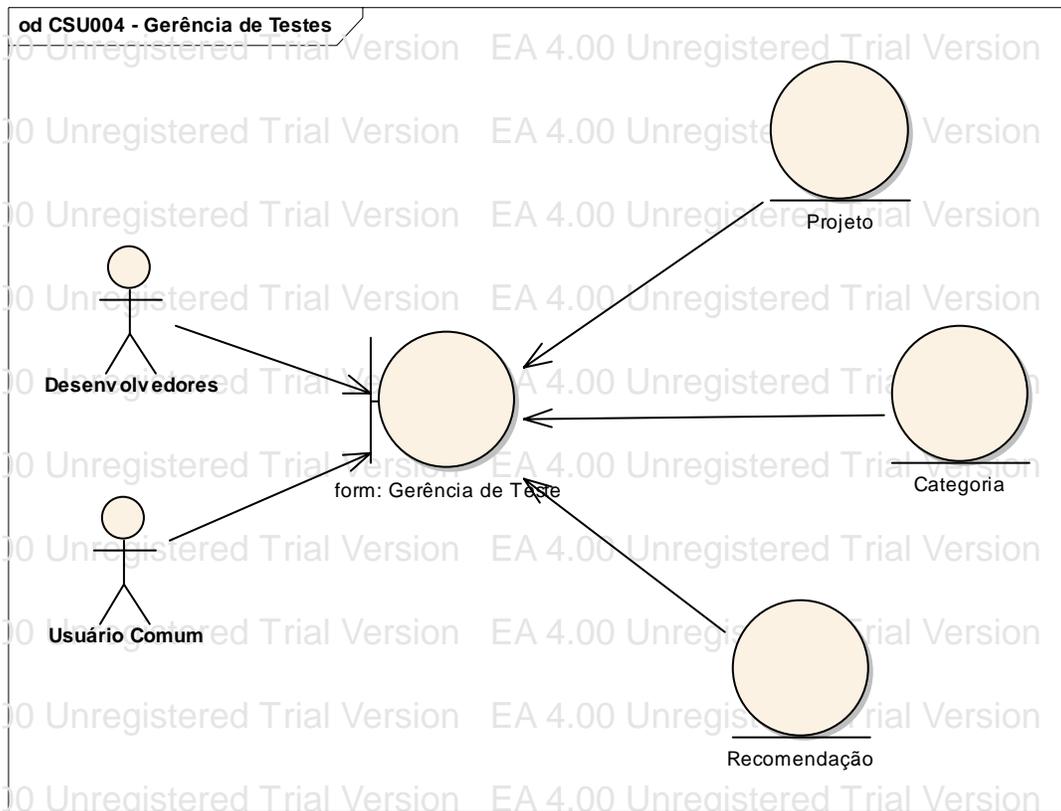


Figura 7: Digrama de Robustez (Gerência de testes)
Fonte: Autor do projeto

O sistema apresenta a tela form: Gerência de Teste, o usuário seleciona na entidade Pprojeto os projetos cadastrados, de acordo com o projeto selecionado é relacionado suas respectivas categorias e cada categoria suas respectivas recomendações.

3.3 Diagrama de atividades

O diagrama de atividades é um diagrama de estados, que representa uma atividade, em vez do estado do objeto. O mesmo pode ser visto como uma extensão de fluxogramas, que possui notação para representar ações concorrentes, juntamente com a sua sincronização. (BEZERRA, 2002).

3.3.1 Gerência de Projeto

Capturados as ações e os resultados obtidos da execução do caso de uso

Gerência de Projeto, obteve-se o diagrama a seguir:

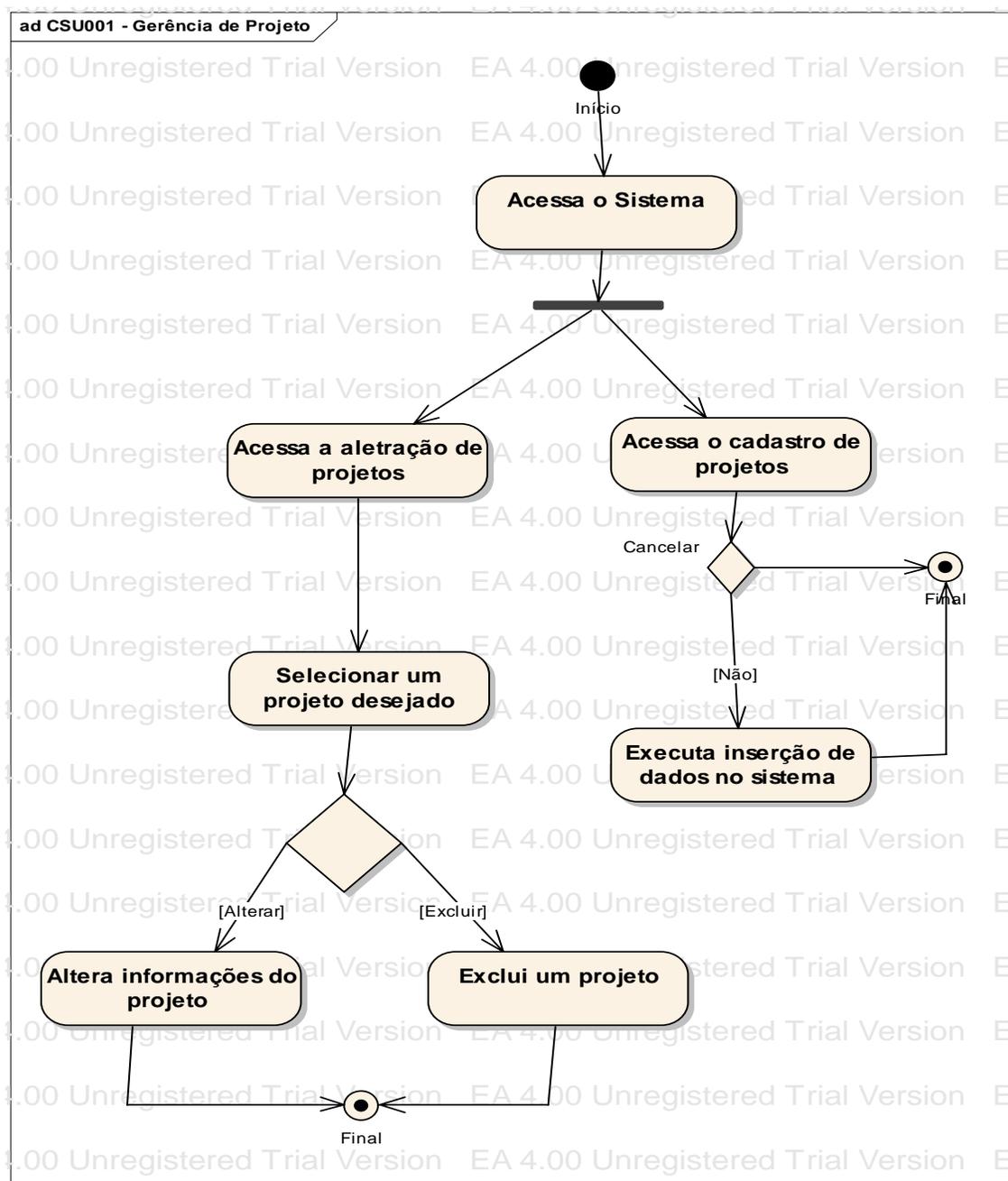


Figura 8: Diagrama de Atividades – Gerência de Projeto
Fonte: Autor do projeto

Iniciando o acesso ao cadastro de projeto pode-se optar por cadastrar novo projeto ou consultar um novo projeto, tendo também a opção de sair. A partir do cadastrar projeto ao finalizar executa o próximo caso de uso. Já pela consulta pode-se optar por fazer alguma alteração e sair ou excluir o projeto consultado e sair do caso de uso.

3.3.2 Gerência de Categorias

Capturadas as ações e os resultados obtidos da execução do caso de uso Gerência de categorias, obteve-se o diagrama a seguir:

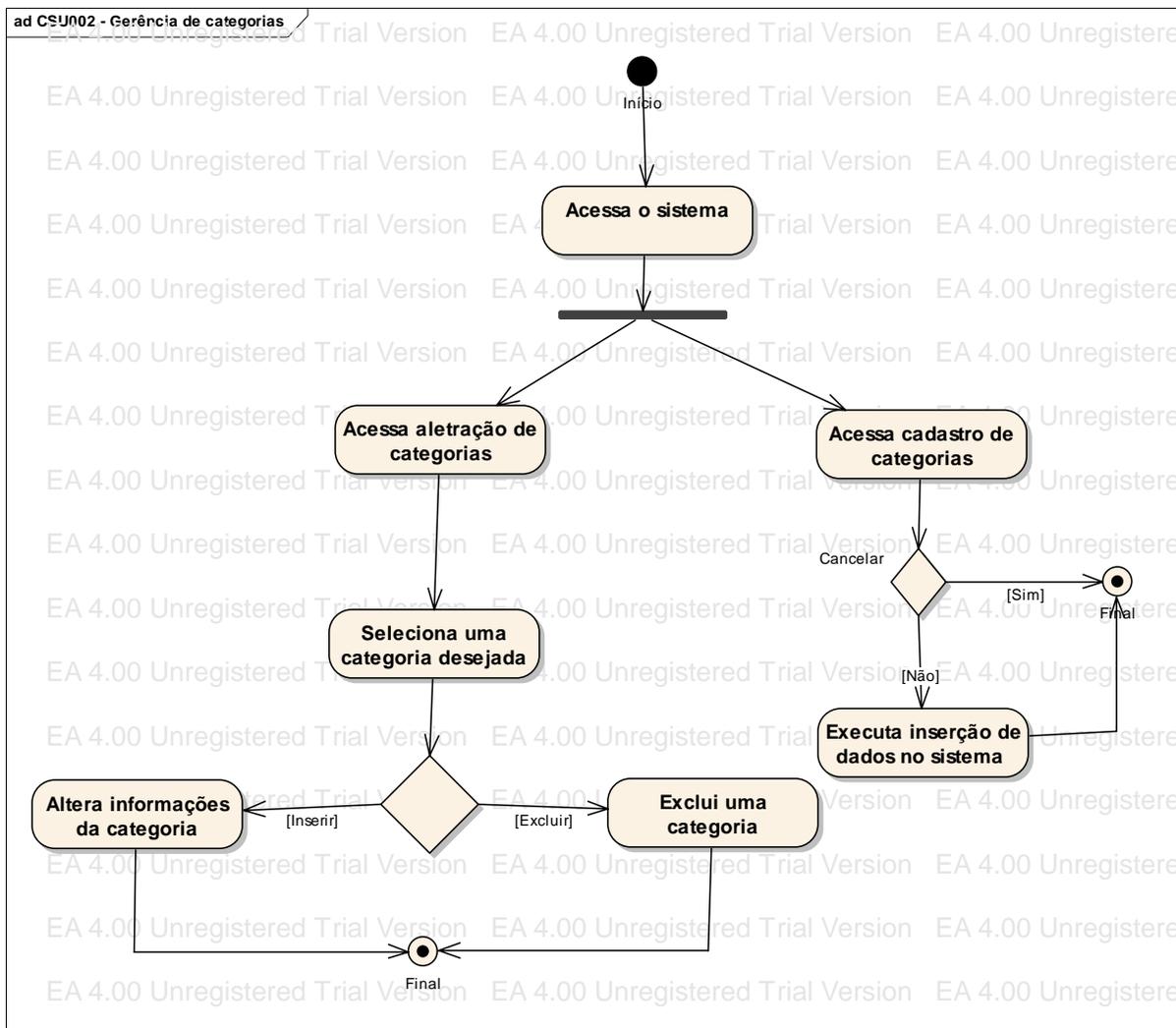


Figura 9: Diagrama de Atividades – Gerência de Categorias

Fonte: Autor do projeto

Iniciando o acesso ao cadastro de categorias pode-se optar por cadastrar uma nova categoria ou consultar uma nova categoria, tendo também a opção de sair. A partir de cadastrar categorias tem a opção de cadastrar quantas categorias o usuário deseja, ao finalizar executa o próximo caso de uso. Já na consulta pode-se optar por fazer alguma alteração e sair ou excluir a categoria consultada e sair do caso de uso.

3.3.3 Gerência de Recomendações

Capturados as ações e os resultados obtidos da execução do caso de uso Gerência de recomendações, obteve-se o diagrama a seguir:

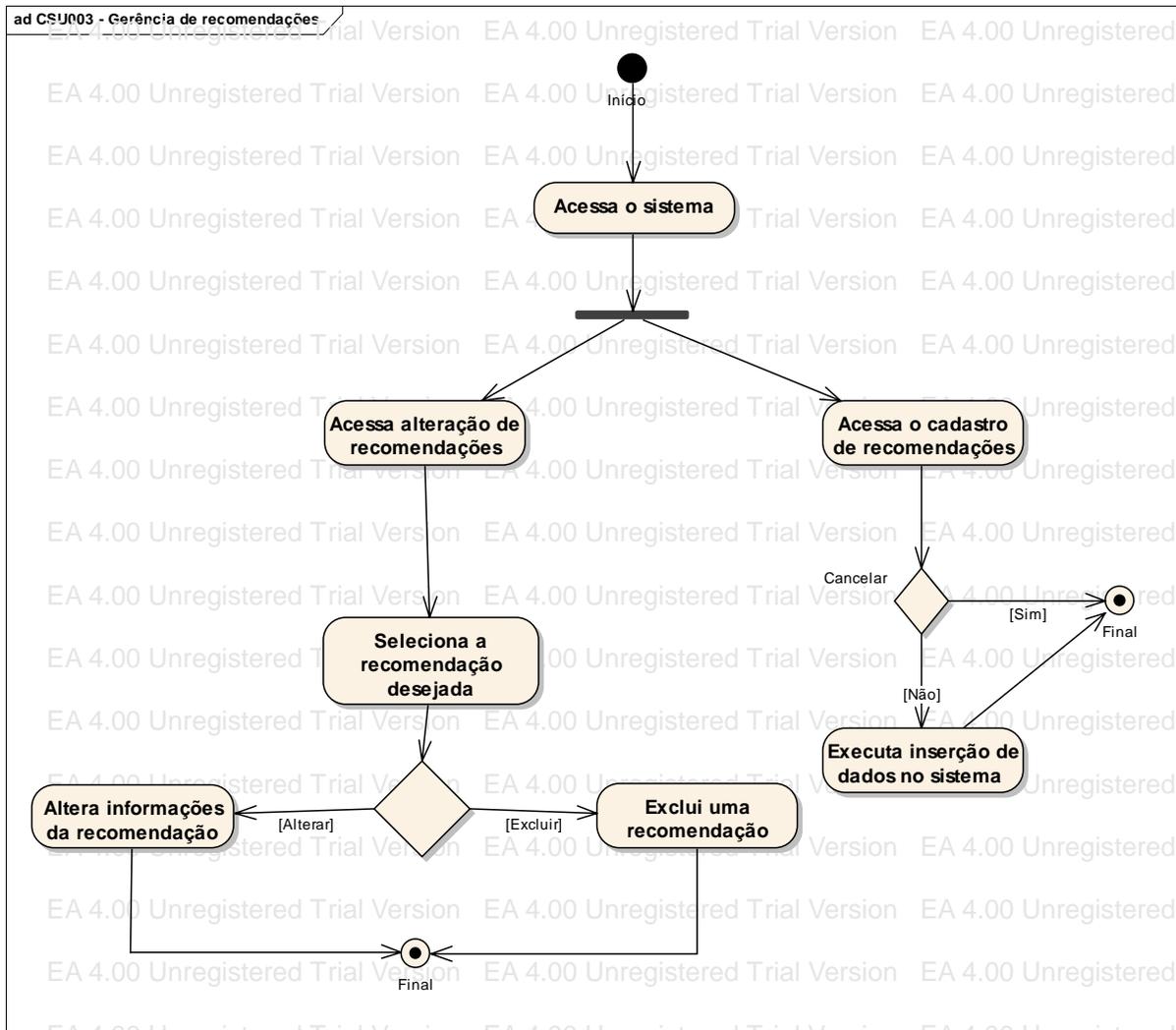


Figura 10: Diagrama de Atividades – Gerência de Recomendações

Fonte: Autor do projeto

Iniciando o acesso ao cadastro de recomendações, pode-se optar por cadastrar uma nova recomendação ou consultar uma recomendação, tendo também a opção de sair. A

partir de cadastrar recomendações tem a opção de cadastrar quantas recomendações o usuário deseja. Na consulta pode-se optar por fazer alguma alteração e sair ou excluir a recomendação consultada e sair do caso de uso.

3.3.4 Gerência de testes

Capturados as ações e os resultados obtidos da execução do caso de uso

Gerência de testes, obteve-se o diagrama a seguir:

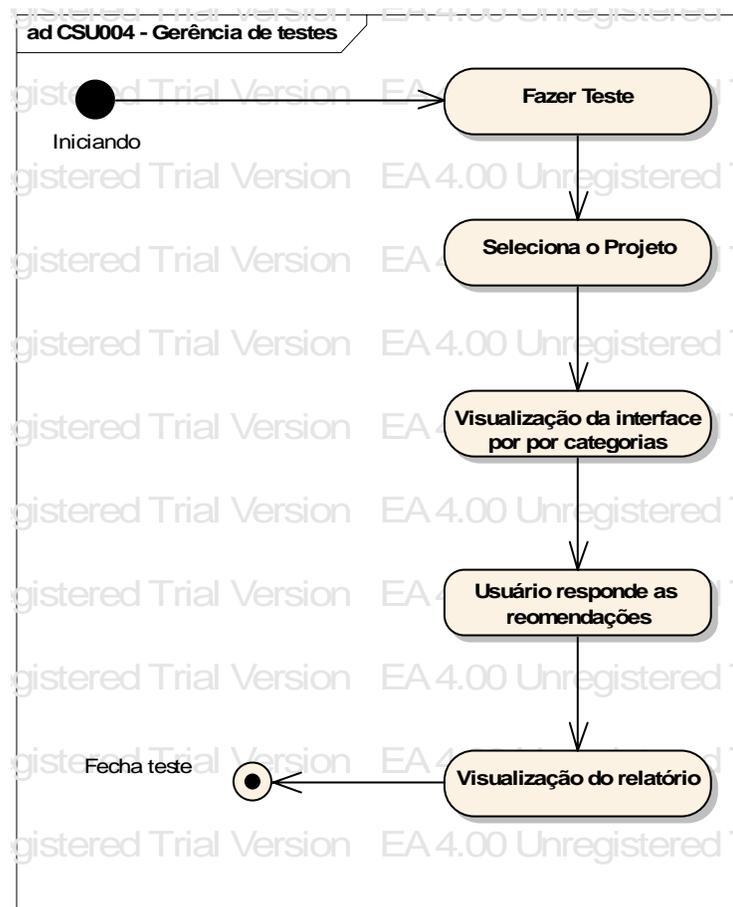


Figura 11: Diagrama de Atividades – Gerência de Testes
Fonte: Autor do projeto

Iniciando o acesso ao fazer teste tem a opção de selecionar um projeto já cadastrado, também tem a opção de sair. Após a visualização da interface o usuário responde as questões apresentadas, e visualiza o relatório, e sai de teste.

3.4 Diagrama de Classes de Interfaces

O diagrama de interfaces permite a visão do relacionamento entre todas as interfaces existentes no sistema.

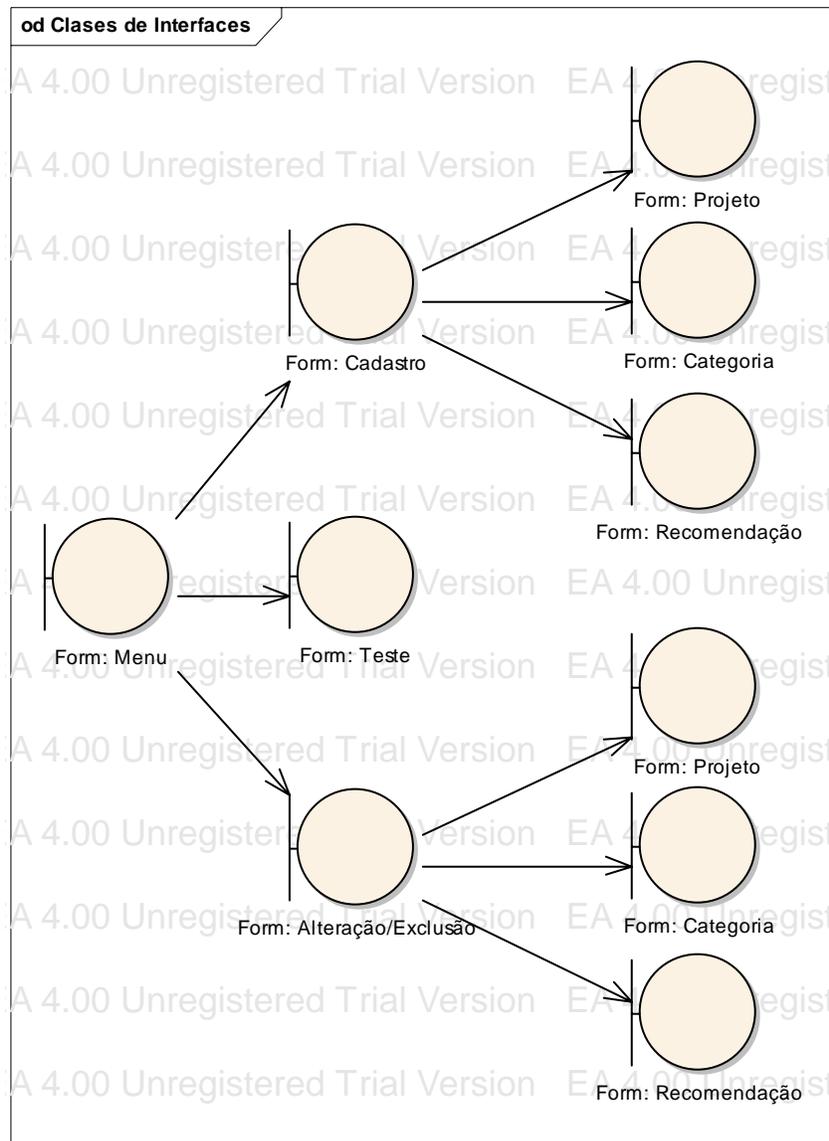


Figura 12: Diagrama de Classes de Interfaces
Fonte: Autor do projeto

3.5 Diagrama de Classes Persistentes

Diagrama de classes persistentes são diagramas que mostram um conjunto de classes importante não só para visualização mas especificação e a documentação de modelos estruturais, e também para construção dos sistemas executáveis por meio de

engenharia de produção e reversa. Diagrama de classes persistentes geralmente é utilizado na modelagem do banco de dados (BOOCH, 2000).

Abaixo e mostrado o diagrama lógico das classes do presente projeto:

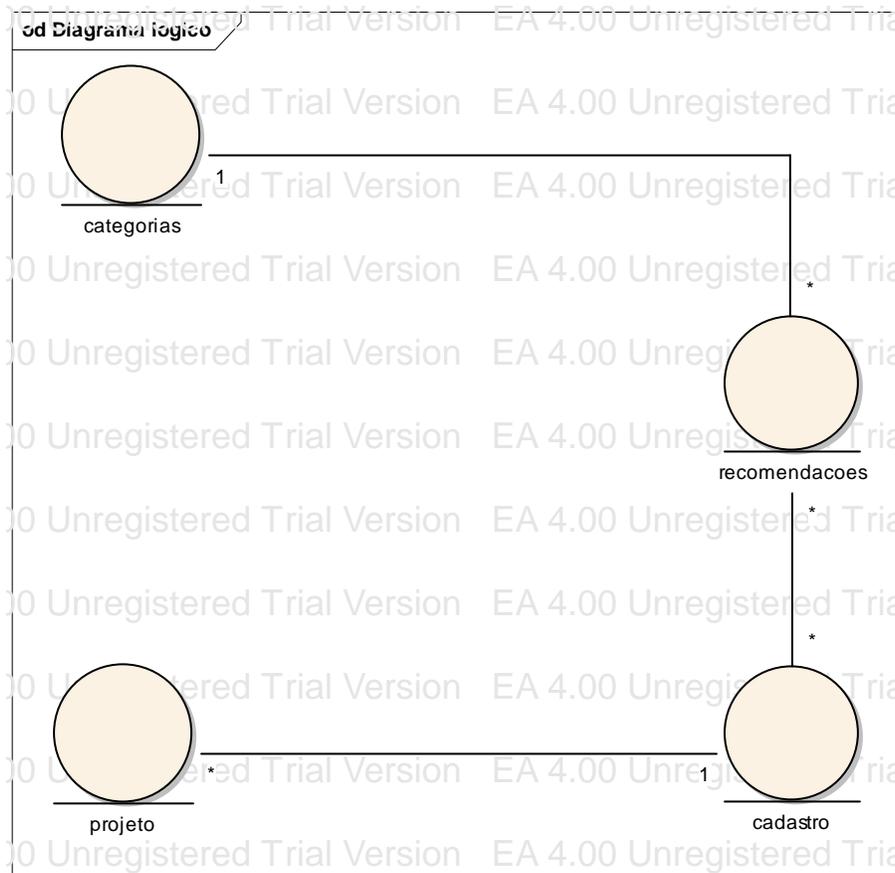


Figura 13: Diagrama de Classes Persistentes
Fonte: Autor do projeto

Abaixo e mostrado o diagrama físico das classes do presente projeto:

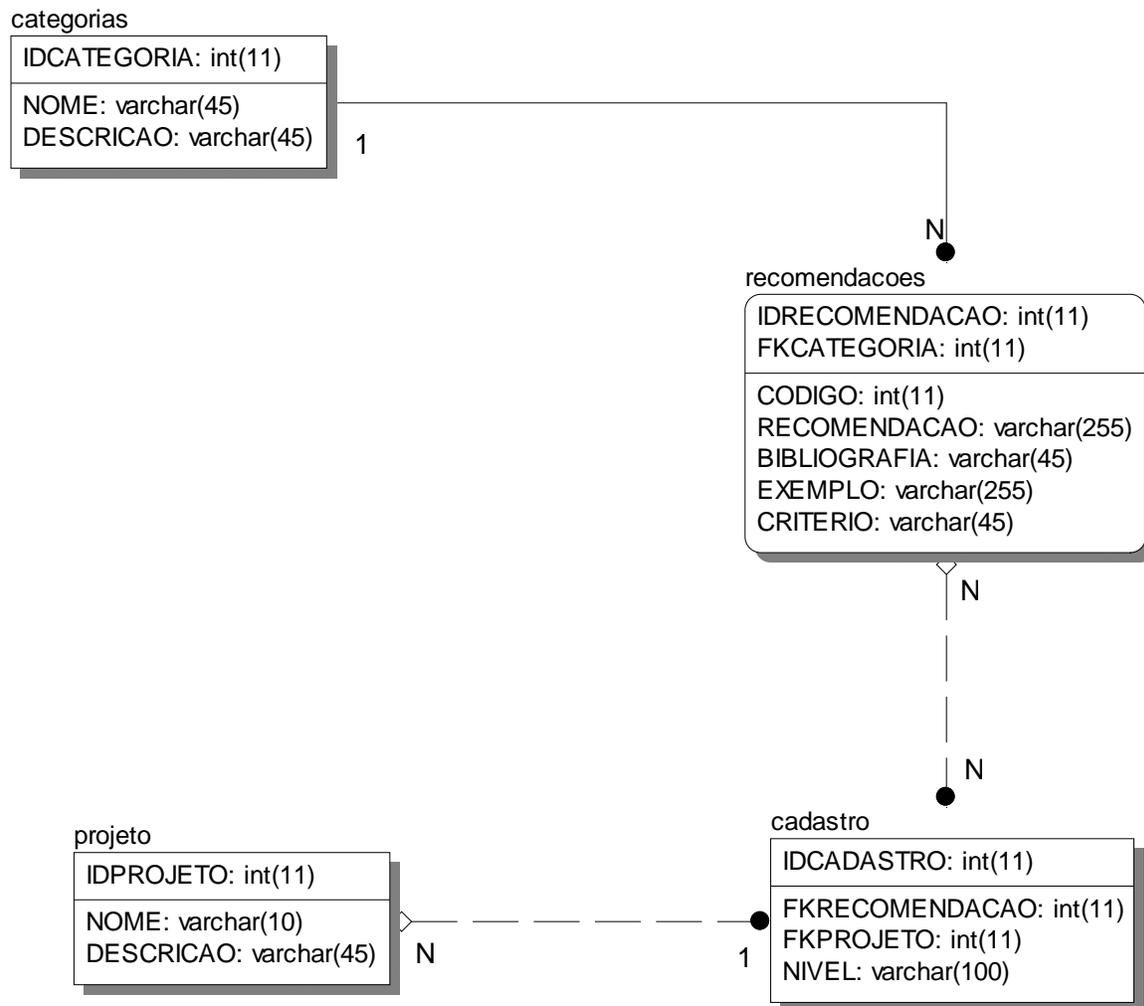


Figura 14: Diagrama de Classes Persistentes
Fonte: Autor do projeto

3.6 Conclusão

A modelagem permite um bom nível de entendimento da solução proposta, podendo prever erros, antes que esses sejam implementados.

A UML é uma linguagem gráfica que proporciona visualização, especificação, construção e documentação de sistemas, proporcionando um maior detalhamento do problema para facilitar a manutenção e agilidade na implementação. São vantagens para realizar um projeto que faz uso da UML.

Segundo Bezerra (2002) um sistema é descrito em certo número de visões, onde cada uma representa uma projeção da descrição completa e mostra aspectos particulares do sistema. Cada visão é mostrada por um diagrama que contém informações que enfatizam os aspectos particulares do software.

Neste projeto foi realizada a modelagem UML, para proporcionar o maior entendimento sobre a aplicação.

4 DESENVOLVIMENTO E VALIDAÇÃO

Neste capítulo são descritas as etapas do processo de desenvolvimento do *Checklist* automatizado, incluindo o processo de coleta de informações e análise para a obtenção dos dados necessários para a implementação deste projeto até as etapas de implementação e validação.

4.1 A Coleta de Informações

No primeiro momento da análise de requisitos, iniciou-se um processo de discussão juntamente com o orientador em torno da possibilidade da realização do projeto.

A idéia do projeto surgiu da necessidade do LPU - Laboratório de Pesquisa em Usabilidade localizado na UNISUL. O laboratório realiza avaliação em usabilidade em empresas parceiras.

A partir das pesquisas realizadas no laboratório, percebeu-se que um *checklist* automatizado ofereceria apoio as avaliações, pois o mesmo realiza avaliações em diferentes áreas de conhecimento.

4.2 Análise de softwares existentes

Uma das etapas do projeto foi à análise do estado da arte de ferramentas disponibilizadas ao público aberto. Entre elas estudou-se o *checklist* oferecido pelo Laboratório de Utilizabilidade da Universidade Federal de Santa Catarina o Labutil.



Figura 15: Tela do sistema Ergolist
Fonte: Labutil (2006)

Observou-se na análise que o referido *checklist* é estático e não varia por área de conhecimento, sendo que o mesmo apesar de gratuito oferece após sua utilização apenas um relatório totalizador de erros e acertos como apresentado abaixo:

Laudo Final	
Presteza	
Total de Questões:	17
Respondidas:	15
Não Respondidas:	2
Questões Conformes:	8
Questões Não conformes:	5
Questões Não Aplicáveis:	2
Questões Adiadas:	0
<hr/>	
Agrupamento por localizacao	
Total de Questões:	11
Respondidas:	11
Não Respondidas:	0
Questões Conformes:	4
Questões Não conformes:	7
Questões Não Aplicáveis:	0
Questões Adiadas:	0
<hr/>	
Total	
Total de Questões:	28
Respondidas:	26
Não Respondidas:	2
Questões Conformes:	12
Questões Não conformes:	12
Questões Não Aplicáveis:	2
Questões Adiadas:	0

Figura 16: Tela do do sistema Ergolist
 Fonte: Labutil (2006)

O conteúdo do *checklist* é bastante genérico e classificado pelos critérios ergonômicos de Bastien e Scapin. A generalidade do mesmo faz com que em muitos casos torne-se ineficiente, pois não é possível selecionar ou adaptar-se as questões ao tipo de interface que será avaliada.

4.3 Histórico do Desenvolvimento

Finalizada a etapa de discussões iniciou-se a modelagem do projeto utilizando-se a linguagem unificada UML descrita na integra no capítulo 3. Para a definição e uso da UML utilizou-se a ferramenta Enterprise Architect versão 4.00.

Após a modelagem partiu-se para a seleção de tecnologias de desenvolvimento. A seqüência de utilização das ferramentas ocorreu conforme apresentado na figura 17.

Utilizou-se o PHP para tornar as páginas dinâmicas na realização de funções de cadastramento, alteração, exclusão, consulta e relatórios de avaliação.

O PHP é uma linguagem de programação de computadores interpretada, livre e muito utilizada para gerar conteúdo dinâmico na Web. É uma linguagem de fácil aprendizado e de uso para scripts dinâmicos. O PHP é uma linguagem orientada à objetos.

A linguagem surgiu por volta de 1994, como um subconjunto de scripts Perl criados por Rasmus Lerdof. Com as adições de Zeev Suraski e Andi Gutmans, dois programadores israelitas pertencentes ao Technion, Instituto Israelita de Tecnologia, que reescreveram o Parser, era lançada em 1997 a PHP 3, hoje já possui a versão 4 e 5. (WIKIPEDIA, 2006).

A linguagem é extremamente modularizada, o que a torna ideal para instalação e uso em servidores Web. É muito parecida em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++ (WIKIPEDIA, 2006).

O PHP significa: *Hypertext Preprocessor*, é uma linguagem de criação embutida em HTML no servidor. Ele tem pouca relação com o *layout* da página, a maior utilização do PHP é invisível para o usuário final, mas o resultado final do PHP é HTML (CONVERSE; PARK, 2001).

O mecanismo de script do PHP pode ser construído no próprio servidor Web, por ele ser um módulo oficial http Apache, tornando a manipulação dos dados mais eficaz e rápida, além disso, o PHP é compatível com várias plataformas (CONVERSE, 2001).

A utilização do Mysql ocorreu para armazenar categorias, recomendações e projetos.

O MySQL é um sistema de gerenciamento de banco de dados (SGBD), que utiliza a linguagem SQL (*Structured Query Language* - Linguagem de Consulta Estruturada) como interface. O MySQL foi criado na Suécia por dois suecos e um finlandês: David Axmark, Allan Larsson e Michael "Monty" Widenius. O desenvolvimento e manutenção empregam aproximadamente 70 profissionais no mundo inteiro, e mais de mil contribuem testando o software, integrando-o a outros produtos, e escrevendo a respeito do mesmo. (WIKIPEDIA, 2006).

O sucesso do MySQL deve-se em grande medida à fácil integração com o PHP incluído, quase que obrigatoriamente, nos pacotes de hospedagem de sites da Internet oferecidos atualmente. Outra grande vantagem é a de ter código aberto e funcionar em quase, todas as plataformas e sistemas operacionais. É reconhecido pelo seu desempenho e robustez e também por ser multi-tarefa e multi-usuário. (WIKIPEDIA, 2006).

O MySQL é um servidor multi-usuário e multi-encadeado permitindo assim que múltiplos usuários trabalhem com os dados ao mesmo tempo, fornecendo acesso rápido aos dados e assegura que somente o usuário autorizado acesse o sistema SGBD. Além disso ele não possui custo, sob uma licença *Open Source*, e pode ser utilizado em muitos sistemas UNIX e Windows (WELLING,2001).

Utilizou-se do *JavaScript* para validação de campos e para tornar os cadastramentos mais dinâmicos, não sendo necessário o *reload* na página para cada inserção.

JavaScript é uma linguagem de programação criada pela *Netscape* - em 1995 -, que a princípio se chamava *LiveScript*, para atender, principalmente, as seguintes necessidades: validação de formulários no lado cliente (programa navegador); interação

com a página. *Javascript* tem sintaxe semelhante a do Java, mas é totalmente diferente no conceito e no uso:

Oferece tipagem dinâmica - tipos de variáveis não são definidos;

É interpretada, ao invés de compilada;

Possui ótimas ferramentas padrão para listagens (como as linguagens de script, de modo geral);

Oferece bom suporte a expressões regulares (característica também comum a linguagens de script).

Sua união com o CSS (*Cascading Style Sheets*) é conhecida como DHTML. Usando o *Javascript*, é possível modificar dinamicamente os estilos dos elementos da página em HTML. (WIKIPEDIA, 2006).

O *Javascript* é uma linguagem interpretada onde o navegador executa cada linha do script como a recebe. Ela é inserida dentro de um programa HTML. Podendo utilizar objetos predefinidos ou então criar novos objetos de acordo com as necessidades (MAKRON BOOKS, 2001).

As ferramentas utilizadas tiveram sua seleção a partir do conhecimento do aluno.

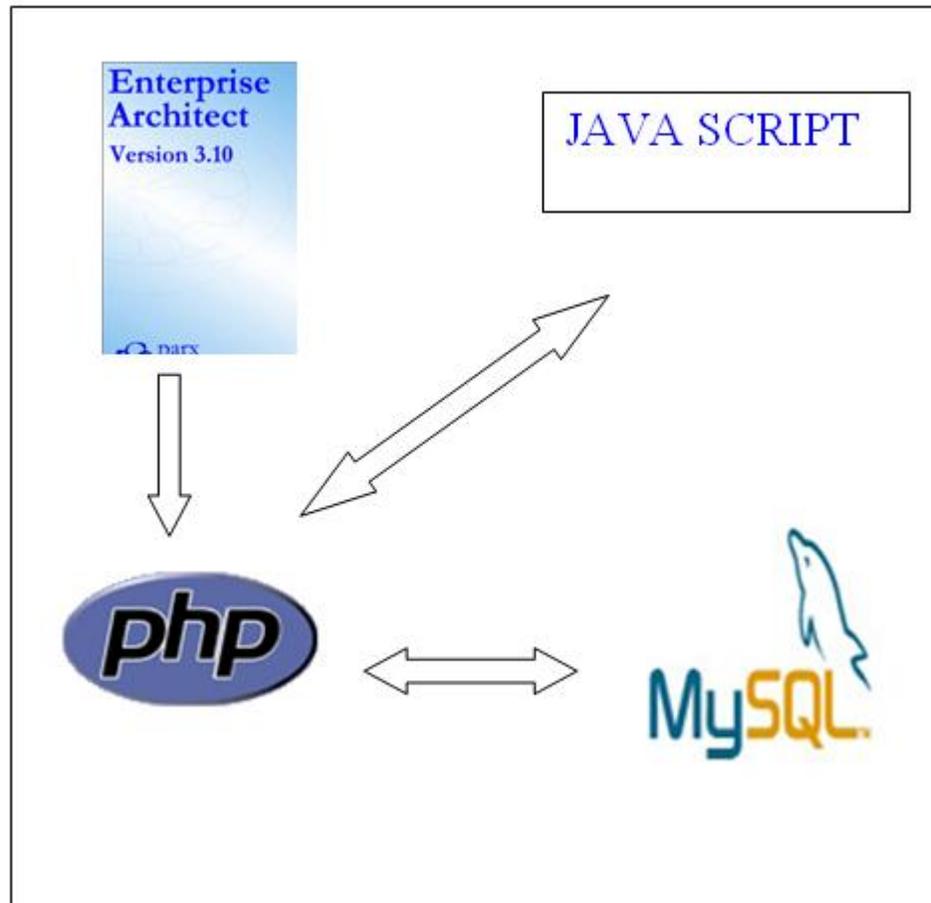


Figura 17: Tecnologias utilizadas
 Fonte: Autor do projeto

O quadro a seguir mostra a versão, o fabricante e a utilização de cada ferramenta utilizada:

Quadro 8: Tecnologias utilizadas
 Fonte: Autor do projeto

Ferramenta	Versão	Fabricante	Utilidade
Enterprise Architect	4.00	Sparx Systems	Ferramenta UML
PHP	4		Programação
MySQL	4.1		Banco de dados
JAVASCRIPT			Programação

4.4 Apresentação do sistema

Serão apresentadas a seguir as seqüências de interação do sistema na forma de telas. A apresentação realizar-se-á na seqüência do menu proposto. A cada tela é apresentado um breve descritivo de suas funções.

A apresentação será realizada a partir de dois módulos principais:

- O módulo de gerência contendo os cadastros e de acesso limitado;
- O módulo avaliador, de acesso irrestrito na Web.

4.4.1 Módulo Gerência

Os cadastros do *checklist* permitem a inclusão, exclusão e alteração de categorias, recomendações e projetos. O acesso de tais cadastros está limitado à equipe do laboratório de usabilidade, não estando disponível ao público em geral.



Figura 18: Página inicial

Fonte: Autor do projeto

A página inicial apresenta o menu lateral com as opções: Categorias, Recomendações, Projetos, Avaliação, Associação projeto/recomendação.

A tela abaixo representa o cadastro de categorias onde cada projeto pode cadastrar categorias de acordo com o domínio de informação do projeto:

The screenshot shows a web application interface titled "Checklist Automatizado para avaliação de interfaces". On the left, there is a vertical navigation menu with the following items: "Categorias", "Recomendações", "Projetos", "Avaliação", "Associação projeto/recomendação", and "Associação projeto/recomendação". The main content area is titled "Cadastro de Categorias" and contains the following elements:

- A dropdown menu labeled "Categorias Cadastradas".
- A text input field labeled "* Nome da Categoria".
- A text input field labeled "Descrição".
- Two buttons: "Incluir" and "Excluir".
- A note at the bottom: "Após fazer as inclusões de categorias clique no botão **Enviar**".

The background of the interface features a blue and green abstract graphic at the top and a logo for "L.P.U. de Pesquisa" and "I.P.U." at the bottom.

Figura 19: Cadastro de categorias
Fonte: Autor do projeto

Nesse cadastro o único campo obrigatório é o nome da categoria. É permitido aos usuários o cadastramento de todas as categorias de seu projeto. A efetivação do cadastro ocorre após o clique no botão “Enviar”.

Checklist Automatizado para avaliação de interfaces

Cadastro de recomendações

* Categorias

Recomendações Cadastradas

* Código

* Recomendação

Exemplo

Bibliografia

Critério

Após fazer as inclusões de recomendações clique no botão

Figura 20: Cadastro de recomendações
Fonte: Autor do projeto

Para realizar o cadastro das recomendações é necessário que haja uma categoria cadastrada, pois as recomendações estão vinculadas à mesma. Os campos obrigatórios do formulário são: Categorias, Código e a Recomendação. Se o campo obrigatório não for preenchido o botão “Incluir” não é ativado. Caso não ocorra a inclusão de nenhuma recomendação o botão “Enviar” não é habilitado. Os demais campos são opcionais para o usuário.

Checklist Automatizado para avaliação de interfaces

Categorias
Recomendações
Projetos
Avaliação
Associação projeto/recomendação

Cadastro de Projetos

* Código

* Nome do Projeto

Descrição

Após digitar o Código e o Nome do projeto clique no botão

LPI de Pesquis
I.P.II

Figura 21: Cadastro de projetos

Fonte: Autor do projeto

No cadastro do projeto é necessário o preenchimento dos campos obrigatórios que são: Código e o Nome do Projeto.

Checklist Automatizado para avaliação de interfaces

Associe a recomendação a um projeto

Projeto: Projeto 1 | Categoria: Formulário

Recomendação	Aplicável ao Projeto	Impacto
A apresentação do valor default do campo é visível quando o campo é apresentado? ⓘ	<input checked="" type="checkbox"/>	Alta
A aproximação do rotulo para o campo usado e consistente em todo o formulário? ⓘ	<input checked="" type="checkbox"/>	Alta
A identificação do formulário está consistente em todas as páginas? ⓘ	<input checked="" type="checkbox"/>	Alta
A identificação do formulário está na área de título da janela? ⓘ	<input checked="" type="checkbox"/>	Alta
A pagina está em um formato que assegura que o formulário esteja inteiramente visível? ⓘ	<input checked="" type="checkbox"/>	Alta
A tecla TAB ou outras formas de mudança de campo estão habilitadas? ⓘ	<input checked="" type="checkbox"/>	Alta
Ao apresentar o formulário o estado atual dos botões binários é mostrado? ⓘ	<input checked="" type="checkbox"/>	Alta
Ao concluir o formulário é emitido um sinal e o mesmo reaparece vazio, ou reaparece numa versão padrão para a entrada de novos? ⓘ	<input checked="" type="checkbox"/>	Alta
Ao identificar o erro o cursor e colocado no primeiro campo que contem erro? ⓘ	<input checked="" type="checkbox"/>	Alta
Ao selecionar o estado dos botões binários e modificado? ⓘ	<input checked="" type="checkbox"/>	Alta

Associar esta página

Figura 22: Associação de recomendações
Fonte: Autor do projeto

Para proporcionar a reutilização de recomendações e categorias cadastradas na base de dados, optou-se pela associação das recomendações com um determinado projeto. Após o cadastro do projeto o usuário é direcionado para essa tela onde seleciona o Projeto desejado e a Categoria desejada. A partir deste momento são apresentadas ao usuário as recomendações daquela categoria selecionada. O usuário decide então por meio do *checkbox* se a recomendação é aplicável ao seu projeto ou não. Após este passo define o impacto que essa recomendação terá para o presente projeto. Tal impacto será utilizado na avaliação, gerando um relatório específico por grau de relevância do atendimento da recomendação.

Definiu-se para o uso do *checklist* uma tabela de impacto sobre a interface conforme o Quadro 9:

Quadro 9: Grau de impacto
Fonte: Autor do projeto

Alto	Tem um alto comprometimento no projeto e a alteração é imprescindível
Médio Alto	Tem um alto comprometimento no projeto e a alteração é recomendada
Médio	Tem um médio comprometimento no projeto e a alteração é recomendada
Médio Baixo	Tem um médio baixo comprometimento no projeto e a alteração é recomendada mas não interfere no uso.
Baixo	Tem um baixo comprometimento no projeto e a alteração não é fundamental.

A alteração ou exclusão de uma categoria é realizada a partir de sua seleção.



Figura 23: Alteração e exclusão de categorias
Fonte: Autor do projeto

Nesta opção são listadas as categorias cadastradas no banco de dados, com limite de dez por tela. Caso existam mais de dez categorias cadastradas apresenta-se um ícone de avanço para a próxima tela. Para a remoção de uma categoria não deve haver nenhuma recomendação vinculada a ela, caso tenha é apresentada uma mensagem de confirmação. Para atualizar uma categoria, deve-se clicar no ícone de atualização apresentando a tela abaixo.



Figura 24: Alteração de categoria
Fonte: Autor do projeto

Para realizar a alteração é necessário preencher os campos obrigatórios e clicar no botão alterar.

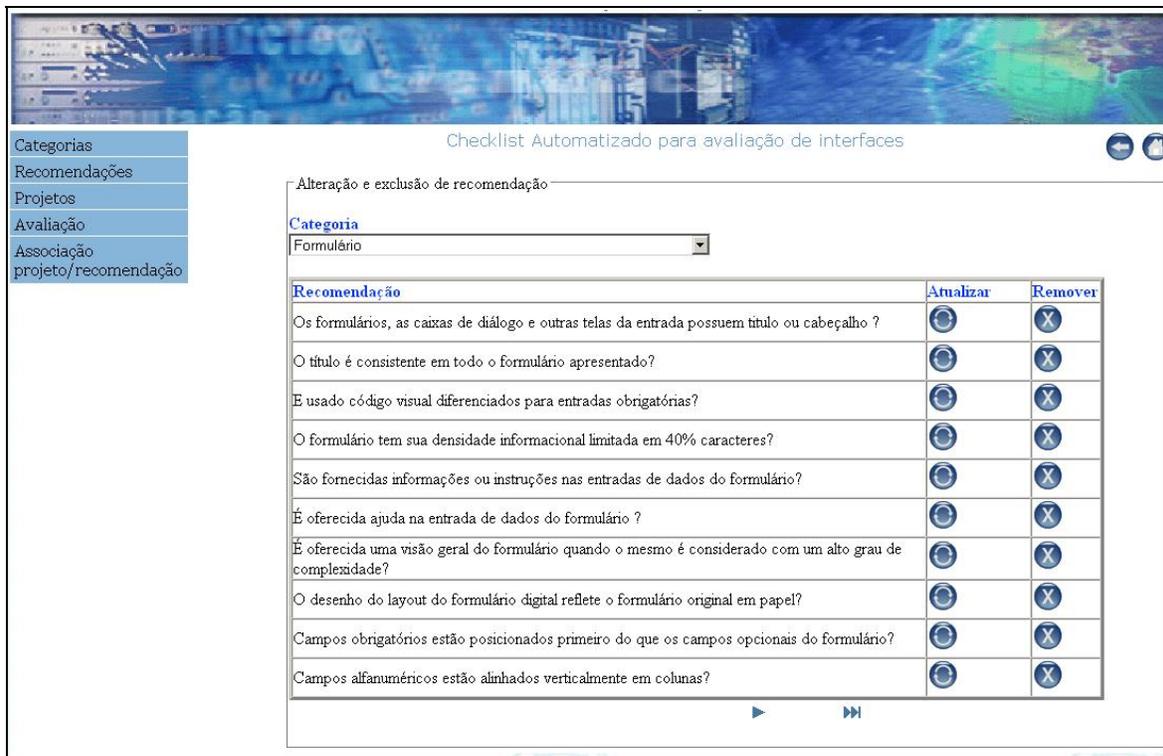


Figura 25: Alteração e exclusão de recomendações

Fonte: Autor do projeto

Para visualizar as recomendações é necessário selecionar uma categoria. Selecionada são apresentadas as recomendações, com limite de dez por tela, vinculadas a essa categoria. Para a remoção de uma recomendação a mesma não pode estar vinculada a nenhum projeto, se estiver é apresentada uma mensagem de confirmação. A atualização da recomendação ocorre após o clique no ícone de atualização.



Figura 26: Alteração de recomendação
Fonte: Autor do projeto

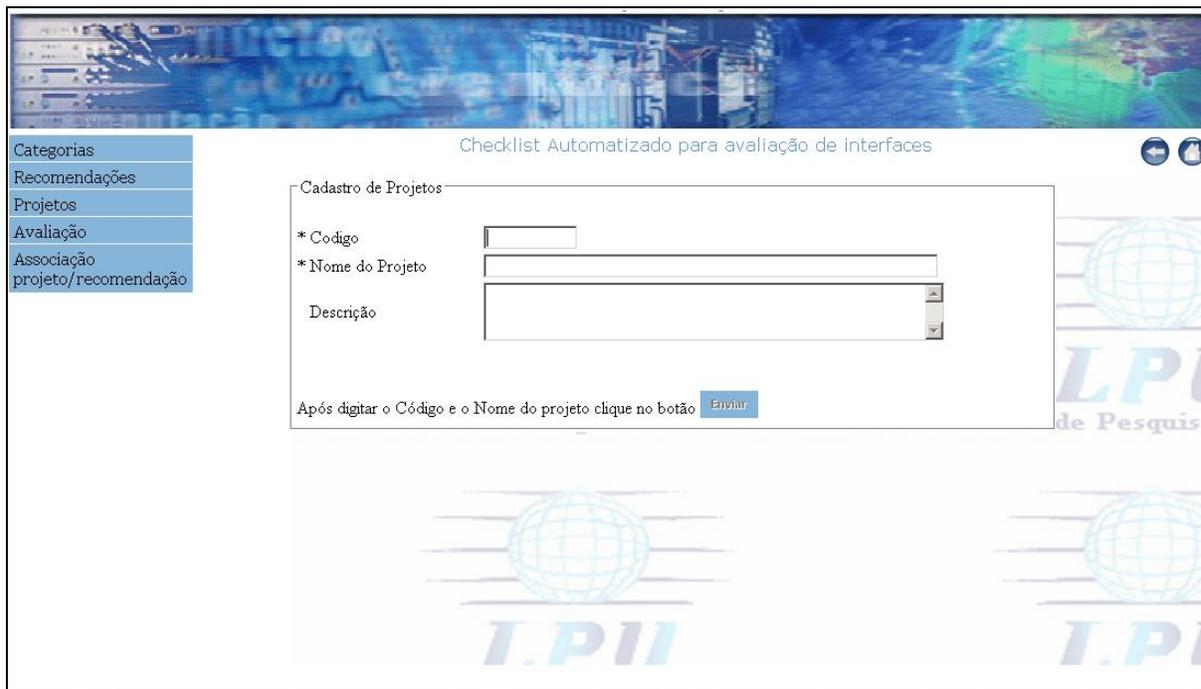
Para realizar a alteração é necessário preencher os campos obrigatórios e clicar no botão alterar.

A exclusão e alteração de projetos ocorrem a partir de sua seleção.



Figura 27: Alteração e exclusão de projetos
Fonte: Autor do projeto

São listadas os projetos cadastrados no banco de dados, com limite de dez por tela. A seguir é apresentada a tela de alteração:



The screenshot shows a web application interface for project management. On the left, there is a vertical navigation menu with the following items: 'Categorias', 'Recomendações', 'Projetos', 'Avaliação', 'Associação projeto/recomendação'. The main content area is titled 'Checklist Automatizado para avaliação de interfaces'. Below this title is a form titled 'Cadastro de Projetos'. The form contains three input fields: '*Codigo' (a small text box), '*Nome do Projeto' (a larger text box), and 'Descrição' (a text area with a vertical scrollbar). At the bottom of the form, there is a blue button labeled 'Enviar' and a note: 'Após digitar o Código e o Nome do projeto clique no botão'. The background of the application features a globe and the text 'LPI de Pesquisa' and 'I.P.I.I'.

Figura 28: Alteração de projeto
Fonte: Autor do projeto

Para realizar a alteração é necessário preencher os campos obrigatórios e clicar no botão alterar.

4.4.2 Módulo Avaliação

O módulo de avaliação está aberto para qualquer usuário, é necessário apenas que o projeto desejado por esse usuário esteja cadastrado na base de dados. O usuário ao acessar o sistema desejando realizar uma avaliação deve selecionar o tipo de projeto,

ambiente de Ensino a Distância, Dispositivos Moveis, etc. No processo de uso do *checklist* foram adotados os seguintes critérios de aceitação quanto à recomendação:

Quadro 10: Critério para avaliação

Fonte: Autor do projeto

Ação	Descrição da Ação
Atende a recomendação	Quando o sistema avaliado atende a recomendação.
Não atende a recomendação	Quando o sistema avaliado não está atendendo a recomendação.
Não aplicável ao projeto	Quando a recomendação não e aplicável para aquele sistema.

Para cada item é possível ao usuário digitar seu comentário.

A tela abaixo apresenta o processo de avaliação:

Checklist Automatizado para avaliação de interfaces

Avaliação

Projeto

Projeto 1

Responda as questões!

Categoria: Formulário	Respostas	Comentários
Os formulários, as caixas de diálogo e outras telas da entrada possuem título ou cabeçalho?	Não aplicável ao projeto	
O título é consistente em todo o formulário apresentado?	Não aplicável ao projeto	
É usado código visual diferenciados para entradas obrigatórias?	Não aplicável ao projeto	
O formulário tem sua densidade informacional limitada em 40% caracteres?	Não aplicável ao projeto	
É oferecida ajuda na entrada de dados do formulário?	Não aplicável ao projeto	
É oferecida uma visão geral do formulário quando o mesmo é considerado com um alto grau de complexidade?	Não aplicável ao projeto	
Campos obrigatórios estão posicionados primeiro do que os campos opcionais do formulário?	Não aplicável ao projeto	
Campos alfanuméricos estão alinhados verticalmente em colunas?	Não aplicável ao projeto	
Campos alfanuméricos estão justificados a esquerda dentro de cada coluna?	Não aplicável ao projeto	
Os campos numéricos estão justificados a direita?	Não aplicável ao projeto	
Rótulos possuem uma diferença significativa de tamanho?	Não aplicável ao projeto	
A identificação do formulário está consistente em todas as páginas?	Não aplicável ao projeto	
Os rótulos de campos de entrada são distintos ou únicos?	Não aplicável ao projeto	
A aproximação do rotulo para o campo usado e consistente em todo o formulário?	Não aplicável ao projeto	

Recomendação: O uso de diferentes dispositivos de entrada de dados e minimizado?

Bibliografia

ISO9241-17

Exemplo

mouse, teclado, leitora, etc.

Critério

Considerações de entrada

Figura 29: Avaliação

Fonte: Autor do projeto

Ao lado da recomendação é apresentado um ícone de informações. Para as recomendações são mostrados três campos:

- **Bibliografia:** Referência bibliográfica da recomendação;
- **Exemplo:** Acrescenta um exemplo positivo ou negativo da aplicação daquela recomendação.
- **Critério:** O que a recomendação está realmente avaliando na interface.

Ao terminar a avaliação o usuário clica no botão enviar e é gerado o relatório para aquela avaliação.

O relatório é gerado em arquivo *pdf*, o relatório não é armazenado pelo projeto, ao gerar um outro relatório o último é substituído., ficando a critério do usuário o salvamento do *pdf*.

A tela abaixo apresenta o relatório:

 Laboratório de Pesquisa em Usabilidade 		
Relatório de conformidades para o projeto: Projeto 1		
----- Resultados de itens não atendidos conforme a recomendação -----		
Recomendação	Comentário	Impacto
Os formulários, as caixas de diálogo e outras telas da entrada possuem título ou cabeçalho ?		Alto
----- Resultados por percentuais da avaliação conforme o impacto -----		
Percentual total de de recomendações com impacto Alto que não atenderam as recomendações 100.00% ;		
Percentual total de de recomendações com impacto Médio Alto que não atenderam as recomendações 0.00% ;		
Percentual total de de recomendações com impacto Médio que não atenderam as recomendações 0.00% ;		
Percentual total de de recomendações com impacto Médio Baixo que não atenderam as recomendações 0.00% ;		
Percentual total de de recomendações com impacto Baixo que não atenderam as recomendações 0.00% .		
----- Resultados pro percentuais da avaliação por conformidade -----		
Percentual total de recomendações aplicáveis ao projeto que atenderam as recomendações 75.00% ;		
Percentual total de recomendações aplicáveis ao projeto que não atenderam as recomendações 25.00% .		

Figura 30: Relatório de avaliação
Fonte: Autor do projeto

É apresentado no relatório o projeto analisado, em seguida são listadas as recomendações não atendidas no projeto, seu respectivo comentário se for digitado, e se grau de impacto. Em seguida são apresentados os resultados em percentual conforme o impacto. Finalizando é apresentado o percentual por conformidade das recomendações atendidas e não atendidas, as que não são aplicáveis ao projeto não são analisadas no relatório.

4.5 Validação

Segundo Sommerville (2003), a validação de software destina-se a descobrir se um sistema está de acordo com as especificações e se atende às expectativas do usuário. Este procedimento verifica por revisão e inspeções, os estágios do desenvolvimento do software, até o usuário final.

O processo de validação do sistema deu-se em duas etapas distintas.

- 1) Validação de usabilidade com o autor e a orientadora;
- 2) Validação das funcionalidades com a equipe do laboratório e outros usuários.

4.5.1 Validação de usabilidade

Durante a validação de usabilidade, tomou-se como referência a experiência do autor do projeto, relacionados à avaliação de interfaces web. Como descrita por Rubin (1994) a técnica utilizada para avaliação do projeto foi à heurística, já que a mesma depende da experiência do avaliador.

Foi verificada a consistência das telas, a prevenção de erros, os rótulos e campos. No primeiro momento foram alterados rótulos e as mensagens de orientação ao usuário para ficarem mais significativas ao usuário. Após essa verificação foram analisadas as cores usadas no site, onde os rótulos foram alterados para preto, melhorando a legibilidade das telas. Os campos de entrada foram aumentados.

A seqüência do menu foi alterada para melhor atender as expectativas do usuário.

4.5.2 Validação das funcionalidades

O presente trabalho foi submetido à equipe do laboratório de pesquisa em usabilidade para utilizá-lo, a partir do *checklist* construído baseando-se na norma ISO9241. As recomendações elaboradas pelo laboratório foram digitadas no Glist. Ao iniciar-se o cadastro percebeu-se que a dimensão do site era pequena para as entradas de dados, como foi utilizado *combobox*, este não apresentava informações maiores, foi necessário então o redimensionamento da página. Campos *single line* foram alterados para *multiple line* para não encobrirem informações digitadas.

Finalizada a digitação das recomendações, a equipe do laboratório utilizou o sistema para realizar as avaliações de uma empresa parceira. Foi utilizado apenas o *checklist* da

parte 17 da norma ISO 9241, foram analisadas 10 telas. A figura 31 apresenta um recorte desta avaliação.



Figura 31: Página inicial, Módulo Avaliação.

Fonte: Autor do projeto

Ao entrar no endereço² o usuário pode visualizar o resumo do trabalho, possui a opção de enviar um e-mail para o autor do projeto, e ainda a opção de avaliação de projetos cadastrados.

Caso o usuário possua permissão para cadastros e alteração ele acessa o link no canto esquerdo da página onde consta acesso restrito.

Acessando a avaliação o processo é o mesmo descrito na figura 29 no presente projeto.

² <http://inf.unisul.br/~lpu/checklist>




Laboratório de Pesquisa em Usabilidade

Relatório de conformidades para o projeto: Projeto 1

Resultados de itens não atendidos conforme a recomendação

Recomendação	Comentário	Impacto
É oferecido um mecanismo de rolagem no campo quando a entrada de dados for maior que o tamanho da caixa?	Campo Adicionar sem mecanismo.	Alto
E usado código visual diferenciados para entradas obrigatórias?	Não diferencia as entrada obrigatórias	Alto
O campo possui um tamanho suficiente para que não ocorra desdobramento?	Adicionar manualmente não possui limite	Alto
O formulário oferece uma diferença visível entre campos de entrada opcionais e obrigatórios?	Não possui uma diferença visível	Alto
O formulário possui a opção voltar?	Não possui opção voltar	Alto

Figura 32: Validação do sistema
Fonte: Autor do projeto




Laboratório de Pesquisa em Usabilidade

Relatório de conformidades para o projeto: Projeto 1

Resultados de itens não atendidos conforme a recomendação

Recomendação	Comentário	Impacto
Possui indicações sobre qual o formato da entrada de dados do campo?	Não possui indicação de qual a entrada	Médio Alto

Resultados por percentuais da avaliação conforme o impacto

Percentual total de de recomendações com impacto Alto que não atenderam as recomendações 62.50% ;
 Percentual total de de recomendações com impacto Médio Alto que não atenderam as recomendações 12.50% ;
 Percentual total de de recomendações com impacto Médio que não atenderam as recomendações 25.00% ;
 Percentual total de de recomendações com impacto Médio Baixo que não atenderam as recomendações 0.00% ;
 Percentual total de de recomendações com impacto Baixo que não atenderam as recomendações 0.00% .

Resultados pro percentuais da avaliação por conformidade

Percentual total de recomendações aplicáveis ao projeto que atenderam as recomendações 88.57% ;
 Percentual total de recomendações aplicáveis ao projeto que não atenderam as recomendações 11.43% .

Relatório gerado em: 02/06/06 19:56:50 Laboratório de pesquisa em usabilidade

Figura 33: Validação do sistema 2
Fonte: Autor do projeto

O uso do *checklist* automatizado durante a avaliação ocorreu sem problemas sendo que o relatório atingiu os resultados esperados. Foram observados todos os quesitos funcionais e totalizadores sendo que o mesmo apresentou os resultados esperados.

4.6 Conclusão

Nesta etapa descreveu a existência de software na área de avaliação de usabilidade, e principalmente as tecnologias necessárias para o funcionamento do Glist.

Foi realizada a apresentação do sistema, na qual foram expostas as telas para permitir o entendimento de como se dá a utilização da ferramenta. Ainda foram descritos problemas encontrados na validação e como foram solucionados.

Por fim, encerra-se com a validação da ferramenta pelo laboratório de pesquisa em usabilidade, comprovando assim que os objetivos apresentados no início do projeto foram atendidos.

5 CONCLUSÃO

Durante o desenvolvimento do trabalho foi uma preocupação constante a obtenção de conhecimento sobre usabilidade e técnicas de avaliação. Percebeu-se que existem várias formas de fazer uma avaliação de usabilidade, qualquer uma dessas avaliações, com suas peculiaridades, resolvem problemas relacionados à usabilidade. Mas raramente são utilizadas em empresas de software, por requerer conhecimento, tempo e custo elevado de execução.

O uso de *checklist*, no entanto é um meio barato e eficiente de capturar problemas de usabilidade. É importante ratificar que sendo viável mesmo que o funcionário não tenha o conhecimento sobre usabilidade. A usabilidade é um elemento crítico para o sucesso no mercado de informática e a realização dos testes de usabilidade é um elemento imprescindível no projeto.

Ao validar o sistema comprovou-se a credibilidade do uso do *checklist* como apoio à avaliação em usabilidade realizada pelo laboratório. Outro ponto a se observar é a democratização de conceitos por meio de disponibilização da ferramenta na internet para avaliações externas.

Além do trabalho realizado para obtenção do título de Bacharel em Sistema de Informação obteve-se durante o ano sua aprovação na forma de artigo no I Seminário de Iniciação Científica da ACADEMIA de Informática sendo este uma oportunidade de divulgação do trabalho realizado.

Importante ainda é salientar a satisfação de saber que o Glist atendeu seus objetivos que está disponível para o uso no endereço <http://inf.unisul.br/~lpu/checklist> A obtenção de conhecimento por parte do aluno em uma área pouco trabalhada durante o curso abre a oportunidade para a especialização em uma área profissional emergente no mercado o Engenheiro em Usabilidade.

Neste momento é tratada a questão da continuidade do trabalho. São sugeridas algumas incorporações:

- Pesquisas de recomendações para alimentação do mesmo, formando a base de dados mais robusta e versátil.
- Incorporar novas funcionalidades, como outros critérios de busca, cadastro e avaliação, a fim de agilizar os processos.
- Oferta do *checklist* no site e a validação de seu uso com usuários externos.

REFERÊNCIAS

BASTIEN, Christian, SCAPIN, Dominique. **Ergonomic criteria for the evaluation of human-computer interfaces**. Racquencourt, France: INRIA, 1993.

BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Campus, 2002.

BOOCH, Grandy. **UML: guia do usuário: o mais avançado tutorial sobre Unified Modeling Language (UML)**. Rio de Janeiro: Campus, 2000.

BROWNE, Dermot P., SUMMERSGILL, Robert, STRADLING, Phil. The user interface: the poor relation in structured methods. In: **Advances in human-computer interaction: volume 3**. Norwood: Ablex, 1992.

CONSTANTINE, Larry. L., LOCKWOOD, Lucy. A. D. **Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design**, New York: ACM Press, 1999.

CONVERSE, Tim., PARK, Joyce. **PHP4 A Bíblia**. Campus: Rio de Janeiro, 2001.

CYBIS, W. A. **Ergonomia de interfaces homem-computador**. LabÚtil, Programa de Pós-graduação em Engenharia de Produção da Universidade Federal de Santa Catarina, Florianópolis, 1999. Apostila. Disponível em <http://www.labutil.inf.ufsc.br/apostila/apostila.htm>, acesso em 15/09/2005.

DE WAAL, M. E., VAN DER HEIDEN, Gerard. H. **The evaluation of user-friendliness in the design process of user interfaces.** Human factors in information systems analysis and design. Elsevier Science Publishers B.V., IFIP, p. 93-103, 1990.

DIAS, Cláudia. **Heurísticas para avaliação de usabilidade de portais corporativos.** Disponível em http://www.geocities.com/claudiaad/heurísticas_web.html. Acessado em 06 de outubro de 2005.

DIAS, Cláudia. **Usabilidade na web: Criando portais mais acessíveis,** Alta Bodas, RS, 2003.

DIAS, Cláudia. **Métodos de avaliação de usabilidade no contexto de portais corporativos:** um estudo de caso no Senado Federal. Brasília: Universidade de Brasília, 2001. 229p.

DIX, Alan. et al. **Human-computer interaction.** Cambridge: University Press, 1993.

ERGOLIST, Disponível em <http://www.labiutil.inf.ufsc.br/ergolist/index.html>. Acessado em 06 de outubro de 2005.

FISHER, G. Human-computer interaction software: lessons learned, challenges ahead. **IEEE Software.** v. 6, n. 1, p. 44-52, 1989

HEEMANN, V. **Avaliação ergonômica de interfaces de bases de dados meio de checklist especializado.** Florianópolis, 1997. Dissertação (Mestrado Engenharia de Produção), Universidade Federal de Santa Catarina.

HIX, Deborah, HARTSON, H. Rex. **Developing user interfaces: ensuring usability through product & process.** John Wiley & Sons, 1993.

ISO 9241 Part 1(1993). **Ergonomic requirements for office work with visual display terminals,** Part 1 General Introduction ; International Standard ISO 9241-1.

LUZZARDI, Paulo Roberto Gomes. **Crítérios de Avaliação de Técnicas de Visualização de Informações.** Universidade Federal do Rio Grande do Sul, UFRGS, Porto Alegre. Doutorado em Ciência da Computação 2003.

MATIAS, M. **Checklist**: uma ferramenta de suporte à avaliação ergonômica de interfaces. Florianópolis, 1995. Dissertação (Mestrado em Engenharia de Produção), Universidade Federal de Santa Catarina.

NIELSEN, Jakob. **Websites**. Disponível em <http://www.useit.com>. Acesso em 15 de setembro de 2005.

NIELSEN, Jakob. **Projetando websites**. Rio de Janeiro: Campus, 2000.

Nielsen, Jakob. **Design Web Usability: The Practice of Simplicity**. New Riders Publish 1999.

NIELSEN, Jakob. **Usability Engineering**. New Jersey: Academic Press, 1993.

Núcleo técnico e Editorial Makron Books. *JavaScript Passo a Passo Lite*. São Paulo, Makron Books, 2001.

OLIVEIRA, Elaine Rosangela de. **Avaliação ergonômica de interfaces da Scielo – Scientific Electronic Library Online**. Florianópolis, 2001. Dissertação (Mestrado) – Universidade Federal de Santa Catarina.

OSF/Motif SYTLE GUIDE. **Guia de estilos para interface com padrão OSF/Motif**, Cambridge, USA, 1990, Open Software Foundation.

PARIZOTTO, Rosamelia. **Guia de Estilos para Serviços de Informação em Ciência e Tecnologia via Web**. Florianópolis, 1997. Dissertação (Mestrado) – Universidade Federal de Santa Catarina.

RUBIN, J. **Handbook of usability testing: how to plan, design, and conduct effective tests**. New York: J. Wiley, 1994.

SILVA, Gleyson Cezar Leme. **Avaliação de usabilidade visando o aumento da interatividade de interfaces de web-sites**: 2004. Disponível em http://geocities.yahoo.com.br/guns_gnr/. Acessado em 02 de outubro de 2005.

SILVEIRA, Mario Cesar. **Checklist ergonômico:** técnica de inspeção de conformidade ergonômica de software interativo voltado à componentes. Florianópolis, 2001. Dissertação (Mestrado) - Universidade Federal de Santa Catarina.

SCAPIN, D. L. **Guide ergonomique de conception des interfaces home-machine.** Rocquencourt, France: INRIA, 1986.

SHNEIDERMAN, Ben. **Designing the user interface** - Strategies for effective humancomputer interaction. 2.ed. Maryland: Addison-Wesley, 1992.

UNIVERSIDADE DO SUL DE SANTA CATARINA. Grupo de Metodologia Científica. Caderno de metodologia: diretrizes para a elaboração e apresentação de trabalhos acadêmicos. 2. ed. rev. Tubarão, 2003. 96 p.

WELLING, Luke., THOMSON, Laura. **PHP E MySQL: Desenvolvimento Web.** Rio de Janeiro: Campus, 2001.

WIKIPEDIA. **Enciclopédia livre.** Disponível em <http://pt.wikipedia.org/wiki>. Acessado em 22 de abril de 2006.

WINDOWS STYLE GUIDE. **The Windows interface guidelines** - *A guide for designing software*, USA, 1995, Microsoft Corporation.