

Universidade Federal de Santa Catarina  
Departamento de Engenharia de Produção e Sistemas  
Programa de Pós-Graduação em Engenharia de Produção

Avaliação Ergonômica e de Qualidade de Software Interativo: Uma  
Contribuição Metodológica Baseada em Técnicas de Preferência Declarada

Tese submetida à Universidade Federal de Santa Catarina para obtenção do  
título de Doutor em Engenharia de Produção.

KARIN CRISTINA SIQUEIRA RAMOS

Orientadora:  
Prof<sup>a</sup> Dr<sup>a</sup> Mirian Buss Gonçalves

Florianópolis

2004

**OS ATRIBUTOS DESEJÁVEIS EM SOFTWARES PARA  
ROTEIRIZAÇÃO DE VEÍCULOS NA VISÃO DA PREFERÊNCIA  
DECLARADA**

**Karin Cristina Siqueira Ramos**

Esta tese foi julgada adequada para a obtenção do título de **DOCTORA EM ENGENHARIA DE PRODUÇÃO** e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Santa Catarina.

---

Prof. Edson Pacheco Paladini, Dr.  
Coordenador do Curso

Banca Examinadora:

---

Prof<sup>a</sup> Mirian Buss Gonçalves, Dr<sup>a</sup>  
Orientadora

---

Prof. Rutsnei Schmitz, Dr  
Moderador

---

Prof<sup>a</sup> Lourdes M. W. de Almeida, Dr<sup>a</sup>  
Examinadora Externa

---

Prof. Antônio Nelson R. da Silva, Dr.  
Examinador Externo

---

Prof. Antônio Galvão Novaes, Dr.  
Membro

---

Prof. Neri dos Santos, Dr.  
Membro

A meu bebê que, pacientemente,  
teve que esperar (mas já está a caminho...).

## AGRADECIMENTOS

Agradeço primeiramente a minha família, por tudo que fizeram e fazem até hoje por mim. Em especial, agradeço a minha mãe, por me ensinar a estudar e me incentivar a isso e a meu pai, por ser meu exemplo de profissional íntegro e competente.

A meu marido Leonardo, por todo seu amor, apoio e paciência em ouvir minhas angústias ou meus relatos detalhados de sucessos. Obrigada por estares sempre a meu lado.

A minha orientadora, Mirian Buss Gonçalves por confiar em minha capacidade de realizar este trabalho, em uma área até então nova para ambas. Agradeço principalmente por suas idéias e ensinamentos, tanto no âmbito profissional quanto pessoal.

Agradeço também aos pesquisadores e profissionais de empresas do ramo da logística que contribuíram diretamente com este trabalho, respondendo a pesquisa de preferência declarada. Em especial agradeço ao professor e colega Rutsnei Schmitz, por sua disposição em ajudar na fase conclusiva do trabalho.

Ao casal Ana Lúcia e Marcelo que, por saberem o que é fazer uma tese, souberam ser amigos para todas as horas.

Por fim, mas não menos importantes, agradeço a minhas “super” amigas: Yara, Ana, Bina, Lili, Inara, Carol, Bíola, Doca e Paula, por sempre levantarem a minha moral.

De tudo fica uma certeza:  
a certeza de que estamos sempre começando;  
a certeza de que precisamos continuar;  
a certeza de que seremos interrompidos antes de terminar.

Portanto devemos fazer:  
da interrupção um novo caminho;  
da queda um passo de dança;  
do medo uma escada;  
do sonho uma ponte;  
da procura um encontro.

Fernando Pessoa

## RESUMO

Nas últimas décadas, o desenvolvimento da informática e a aparição de novas tecnologias, têm levado a uma proliferação dos softwares; todavia, existem softwares que são colocados à disposição do público sem a preocupação de uma análise mais detalhada de seus concebedores, o que pode vir a comprometer a qualidade ou mesmo a eficiência do produto. Convém, nesses casos, a utilização de um instrumento que auxilie na tomada de decisão quanto à adoção ou não de um certo produto, tornando possível identificar se o mesmo está de acordo com os objetivos almejados ou se é possível adaptá-lo a seu caso específico.

Sob essa ótica, propõe-se no presente trabalho uma metodologia que possa auxiliar na avaliação e tomada de decisão quanto à adoção de um software. Essa metodologia tem como base a ergonomia de software e os estudos da engenharia de software referentes à qualidade. A ergonomia de software oferece bases teóricas e metodológicas para a realização de avaliações e seleções de softwares, estudando as dificuldades relativas à relação existente entre o homem e a máquina, e usando sempre o equilíbrio entre conforto, segurança e eficiência do usuário no uso de produtos informatizados. Da mesma forma, é possível encontrar diversas normas e diretrizes de qualidade de software que podem indicar diferentes critérios a serem considerados no processo de avaliação dos mesmos.

O diferencial introduzido na metodologia aqui proposta e que a difere de outros procedimentos tradicionais de análise ergonômica ou da qualidade, é a incorporação de aspectos comportamentais na avaliação. Essas informações podem ser obtidas a partir do uso de técnicas de preferência declarada, onde entrevistas são realizadas com usuários em potencial, visando identificar suas expectativas em relação aos atributos que o software deve contemplar.

O ferramental desenvolvido é testado em softwares para roteirização de veículos. Inicialmente são identificados os atributos essenciais para esse tipo de produto; na seqüência são realizadas entrevistas com especialistas e empresas do ramo de transporte e logística. A partir dos resultados obtidos é possível identificar o grau de importância de cada um dos atributos considerados, tornando possível inclusive a quantificação da conformidade do software. Os resultados encontrados a partir de tal aplicação vêm a ser uma contribuição bastante significativa para a área de estudos.

## ABSTRACT

In the last decades, the development of the information technology as well as the appearance of new technologies have resulted in software proliferation; however, there are software that are placed at public's disposal with no concern as regards a more detailed analysis of their conceivers. That may come to compromise the product quality or even its efficiency. In these cases, it is convenient to make use of a tool to assist the decision making process as for the adoption or not of a certain product. Therefore, it is possible to identify whether such a product is in agreement with the desired objectives or whether it is possible to adapt it to a specific case.

Under this point of view, it is proposed through the present work a methodology that can assist the evaluation and the decision making process as for the adoption of a software. This methodology is based on the software ergonomics and the studies in software engineering regarding quality. The software ergonomics offers theoretical and methodological basis to the process of evaluating and selecting software by studying the relative difficulties of the existing relation between human being and machine and always using the balance among comfort, security and user efficiency as using computerized products. In the same way, it is possible to find out several norms and directives of software quality that can indicate different criteria to be considered in their evaluation process.

The differential aspect introduced into the methodology proposed here that makes it different from other traditional ergonomic analysis or quality procedures is the incorporation of behavior aspects in the evaluation. These information can be obtained from the use of stated preference techniques where interviews with potential users are carried out aiming to identify their expectation relating to the attributes that the software must contemplate.

The developed tool is tested in software for vehicles routing. Primarily, the essential attributes for this kind of product is identified; following, interviews with specialists and companies of the logistic and transportation field are carried out. From the results obtained, it is possible to identify the degree of importance of each one of the considered attributes making even possible a quantification of the software conformity. The results encountered from such application come to be a very significant contribution to the field of studies.

## SUMÁRIO

RESUMO.....	6
ABSTRACT.....	7
LISTA DE TABELAS .....	12
LISTA DE FIGURAS E QUADROS .....	13
CAPÍTULO 1 – INTRODUÇÃO .....	14
1.1. Contextualização.....	14
1.2. Objetivo Geral.....	16
1.3. Objetivos Específicos .....	17
1.4. Procedimentos Metodológicos.....	18
1.5. Estrutura do Trabalho.....	19
CAPÍTULO 2 – ERGONOMIA DE SOFTWARE.....	21
2.1. Introdução.....	21
2.2. Ergonomia de Software .....	23
2.3. Metodologias para Concepção e Avaliação de Sistemas Interativos .....	28
2.3.1. Critérios Ergonômicos de Scapin e Bastien.....	28
2.3.1.1. Condução.....	29
2.3.1.2. Carga de Trabalho.....	31
2.3.1.3. Controle Explícito.....	34
2.3.1.4. Adaptabilidade.....	36
2.3.1.5. Gestão de Erros.....	36
2.3.1.6. Consistência *.....	39
2.3.1.7. Significado dos Códigos *.....	39
2.3.1.8. Compatibilidade * .....	40
2.3.2. Diagnóstico de Software Interativo Segundo Barthelet .....	41
2.3.2.1. Recomendações ergonômicas para os parâmetros da interface.....	43
2.3.2.2. Diagnóstico de um Software Interativo .....	47
2.3.3. Método Iterativo de Concepção de Valentin, Vallery e Luongsang.....	50
2.3.3.1. Reconhecimento do software .....	52
2.3.3.2. Definição da população e dos cenários.....	53
2.3.3.3. Coleta de dados.....	55



2.3.3.4. Análise dos dados coletados.....	56
2.3.3.5. Síntese dos Resultados .....	56
<b>CAPÍTULO 3 – QUALIDADE DE SOFTWARE E AS NORMAS ISO/NBR.</b>	<b>58</b>
3.1. Introdução .....	58
3.2. Qualidade de Software .....	60
3.3. Normas ISO e NBR para Avaliação da Qualidade de Produtos de Software.....	62
3.3.1. A Norma NBR 13596 .....	62
3.3.1.1. Funcionalidade .....	63
3.3.1.2. Confiabilidade .....	64
3.3.1.3. Usabilidade.....	64
3.3.1.4. Eficiência.....	65
3.3.1.5. Manutenibilidade .....	65
3.3.1.6. Portabilidade.....	66
3.3.2. A Norma NBR ISO/IEC 14598.....	68
3.3.2.1. Estabelecer requisitos de avaliação.....	70
3.3.2.2. Especificar a avaliação.....	72
3.3.2.3. Projetar a avaliação .....	73
3.3.2.4. Executar a avaliação .....	73
3.3.3. A Norma NBR ISO/IEC 12119.....	74
3.3.3.1. Descrição de produto .....	74
3.3.3.2. Documentação de usuário .....	76
3.3.3.3. Programas e dados .....	76
3.3.4. A Norma ISO 9241.....	78
3.3.4.1. Recomendações quanto à adequação da tarefa.....	79
3.3.4.2. Recomendações quanto à auto-descrição.....	79
3.3.4.3. Recomendações quanto à controlabilidade .....	80
3.3.4.4. Recomendações quanto à conformidade com as expectativas do usuário .....	81
3.3.4.5. Recomendações quanto à tolerância a erros.....	81
3.3.4.6. Recomendações quanto à adequação à individualização.....	82
3.3.4.7. Recomendações quanto à adequação ao aprendizado.....	82
<b>CAPÍTULO 4 – TÉCNICAS DE PREFERÊNCIA DECLARADA.....</b>	<b>84</b>
4.1. Introdução .....	84
4.2. Conceituação da Preferência Declarada .....	86

4.3. O Experimento de Preferência Declarada .....	88
4.3.1. Forma e Complexidade do Experimento .....	89
4.3.2. Medição da Escolha.....	90
4.3.3. Escolha do Método de Entrevistas .....	90
4.3.4. Seleção da Amostra .....	91
4.3.5. Análise dos Dados .....	92
4.4. A Função Utilidade .....	93
<b>CAPÍTULO 5 – METODOLOGIA DE AVALIAÇÃO PROPOSTA .....</b>	<b>99</b>
5.1. Introdução.....	99
5.2. Elaboração da Metodologia Proposta.....	100
5.2.1. Etapa de Análise.....	100
5.2.2. Etapa de Concepção.....	101
5.2.2.1. Etapa de Concepção do Módulo 1 .....	101
5.2.2.2. Etapa de Concepção do Módulo 2.....	102
5.2.3. Etapa de Valoração.....	103
5.2.3.1. Etapa de Valoração do Módulo 1 .....	103
5.2.3.2. Etapa de Valoração do Módulo 2 .....	105
5.2.4. Etapa de Revisão Final .....	107
5.3. Estrutura da Metodologia .....	107
5.3.1. Estrutura do Módulo 1 .....	107
5.3.2. Estrutura do Módulo 2.....	114
<b>CAPÍTULO 6 - APLICAÇÃO DA METODOLOGIA DE AVALIAÇÃO.....</b>	<b>117</b>
6.1. Introdução.....	117
6.2. O Contexto da Aplicação.....	117
6.2.1. Softwares de Roteirização.....	119
6.2.1.1. Evolução.....	119
6.2.1.2. Estrutura de um Software de Roteirização.....	121
6.2.1.3. Requisitos de um Roteirizador .....	123
6.3. O Projeto Experimental .....	126
6.3.1. Delineamento Experimental do Conjunto 1.....	126
6.3.2. Delineamento Experimental do Conjunto 2.....	129
6.3.3. Delineamento Experimental do Conjunto 3.....	131
6.3.4. Delineamento Experimental Geral .....	134

6.4. Realização das Entrevistas.....	135
6.5. Calibração dos Parâmetros das Funções Utilidade.....	136
6.6. Exemplo de Aplicação da Metodologia para Avaliação de um Software Roteirizador .....	139
<b>CAPÍTULO 7 – CONCLUSÕES E RECOMENDAÇÕES .....</b>	<b>142</b>
7.1. Contribuições Gerais .....	142
7.2. Contribuições Específicas da Nova Metodologia .....	143
7.3. Contribuições Específicas da Aplicação da Metodologia Proposta.....	145
7.4. Sugestões para Pesquisas Futuras .....	146
<b>REFERÊNCIAS.....</b>	<b>148</b>
<b>APÊNDICES .....</b>	<b>155</b>
<b>ANEXOS .....</b>	<b>164</b>

## LISTA DE TABELAS

Tabela 2.1 – Recomendações para a Condução.....	32
Tabela 2.2 – Recomendações para a Carga de Trabalho.....	34
Tabela 2.3 – Recomendações para o Controle Explícito do Usuário.....	35
Tabela 2.4 – Recomendações para a Adaptabilidade.....	37
Tabela 2.5 – Recomendações para a Gestão de Erros.....	38
Tabela 2.6 – Recomendações para a Consistência.....	39
Tabela 2.7 – Recomendações para o Significado dos Códigos.....	39
Tabela 2.8 – Recomendações para a Compatibilidade.....	40
Tabela 2.9 – Síntese do quadro de análise de um software (Barthet, 1988).....	48
Tabela 3.1 – Características e sub-características da NBR 13596 (Barreto, 2000).....	67
Tabela 6.1 – Níveis dos atributos do Conjunto 1.....	127
Tabela 6.2 – Alternativas e níveis dos atributos do Conjunto 1.....	128
Tabela 6.3 – Formação dos Blocos de Alternativas para o Conjunto 1.....	129
Tabela 6.4 - Níveis dos atributos do Conjunto 2.....	130
Tabela 6.5 – Alternativas e níveis dos atributos do Conjunto 2.....	131
Tabela 6.6 – Formação dos Blocos de Alternativas para o Conjunto 2.....	131
Tabela 6.7 - Níveis dos atributos do Conjunto 3.....	132
Tabela 6.8 – Alternativas e níveis dos atributos do Conjunto 3.....	133
Tabela 6.9 – Formação dos Blocos de Alternativas para o Conjunto 3.....	133
Tabela 6.10 - Níveis dos atributos do Conjunto 4.....	134
Tabela 6.11 – Alternativas e níveis dos atributos do Conjunto 4.....	135
Tabela 6.12 – Resultados da calibração dos parâmetros do Conjunto 1.....	136
Tabela 6.13 – Resultados da calibração dos parâmetros do Conjunto 2.....	137
Tabela 6.14 – Resultados da calibração dos parâmetros do Conjunto 3.....	137
Tabela 6.15 – Resultados da calibração dos parâmetros do Conjunto 4.....	137

## LISTA DE FIGURAS E QUADROS

Quadro 2.1 – Critérios ergonômicos de Scapin e Bastien.....	29
Figura 2.1 – Aportes da psicologia cognitiva e da ergonomia em relação às representações conceitual e externa (Barthet, 1988).....	43
Figura 2.2 – Esquema geral do método (Valentin <i>et. al.</i> , 1993) .....	51
Figura 3.1 – Processo de Avaliação de acordo com a norma ISO/IEC 14598-1 (2001) .....	70
Figura 5.1 – Estrutura da metodologia proposta.....	108

## **CAPÍTULO 1 – INTRODUÇÃO**

### **1.1. Contextualização**

Nas últimas décadas, a sociedade vem sofrendo profundas transformações, principalmente no que diz respeito à revolução tecnológica. Esta veio incrementar atividades, implementar outras até então impossíveis, e, inclusive, substituir a atuação de humanos em outras dessas. Para Teodoro (1992, p.12)

os tempos atuais são de profunda transformação tecnológica originada pela rápida evolução e difusão de novas tecnologias, em particular as associadas aos computadores e às comunicações. Estas tecnologias estão a alterar significativamente os processos de produção de bens materiais, mas também os processos de difusão de idéias e, conseqüentemente, os modos de viver em sociedade.

Essa rápida expansão das tecnologias, principalmente de comunicação e informação, tem levado a um aumento considerável do número de softwares disponíveis no mercado. Todavia, existem softwares que são colocados à disposição do público sem a preocupação de uma avaliação, seja do ponto de vista ergonômico ou mesmo funcional, o que pode levar, por exemplo, a erros na entrada de dados, problemas de processamento das informações, além de uso inadequado de cores e fontes. Problemas desse tipo comprometem a qualidade do produto, podendo isso, inclusive, torná-lo obsoleto.

Dessa forma, surge o seguinte questionamento: Como escolher, dentre algumas opções disponíveis, um software que satisfaça certas necessidades específicas?

Muitas vezes a seleção de um software acaba se dando entre aqueles mais diretamente acessíveis, massivamente divulgados ou, na pior das hipóteses, única e exclusivamente pelo critério financeiro, isto é, é selecionado aquele que despense menor recurso na aquisição.

A seleção de um software interativo implica obrigatoriamente em uma definição prévia dos critérios específicos para avaliar sua qualidade e adequabilidade. Para tanto é necessário conhecer melhor o usuário em potencial do produto e qual tarefa o mesmo deseja implementar com o uso desse. Além disso, faz-se necessária a adoção de um método de

avaliação capaz de considerar tanto as características funcionais quanto à “amigabilidade” da interface (RAY, 1994). Isso porque

a interface do sistema com o usuário é frequentemente o padrão de comparação pelo qual o sistema é julgado. Uma interface difícil de usar, na melhor das hipóteses, irá resultar em um alto nível de erros, na pior será descartado, independente de sua funcionalidade. Se a informação é apresentada de modo confuso e desorientado, o usuário pode não entender o significado da informação. Ele pode iniciar uma seqüência de ações que corrompa dados ou mesmo cause falha geral no sistema (SOMMERVILLE, 1997, p. 320 – tradução livre).

Porém, segundo Shneiderman (1998, p. 10)

quando um sistema interativo é bem desenhado, a interface quase desaparece, permitindo ao usuário se concentrar em seu trabalho, exploração ou prazer (...) criar esse ambiente onde as tarefas transcorrem ao menor esforço e o usuário flui, requer um grande e árduo trabalho ao designer (tradução livre).

A ergonomia, mais especificamente a ergonomia de software, oferece bases teóricas e metodológicas para enfrentar o problema da avaliação e seleção de softwares, dado que estuda as dificuldades relativas à relação existente entre o homem e a máquina, usando sempre o equilíbrio entre conforto, segurança e eficiência do usuário no uso de produtos informatizados.

Da mesma forma, já existem atualmente diversas normas e diretrizes que podem indicar diferentes critérios a serem considerados no processo de avaliação da qualidade de software. Essas fazem especial consideração a questões de programação, design e documentação, podendo analisar critérios tanto quantitativa como qualitativamente.

Considerando que um mesmo software pode ser manipulado por diferentes pessoas, é importante conhecer as habilidades físicas, cognitivas e perceptivas desses usuários, fazendo considerações, inclusive, quanto a diferenças individuais e à diversidade cultural. Além disso, é importante entender que diferentes pessoas têm diferentes objetivos a alcançar com o uso de um software, portanto, cada uma delas terá uma avaliação particular dessa experiência. Assim sendo, é extremamente importante, na seleção de um produto informatizado, levar em consideração a opinião dos futuros usuários do mesmo.

Sendo assim, surge a seguinte pergunta: Que critérios considerar para desenvolver uma metodologia de avaliação de software interativo, considerando tanto aspectos ergonômicos e de qualidade quanto as preferências dos futuros usuários?

Um dos métodos que tem se mostrado bastante eficaz na obtenção de informações relevantes dos usuários é a adoção de técnicas de preferência declarada. Estas, utilizadas inicialmente em pesquisas de *marketing*, buscam identificar, através de entrevistas, quais atributos o usuário considera mais importantes dada uma certa situação. Para tanto, é apresentado a ele um conjunto de opções, reais ou hipotéticas, que o mesmo deve selecionar, ordenar ou pontuar de acordo com suas preferências. A partir das respostas obtidas, é possível estabelecer um valor de utilidade, que nada mais é do que a medida da atratividade do bem ou serviço em termos de seus atributos. A função utilidade de uma alternativa tem como objetivo decompor as preferências admitidas dos entrevistados em partes de utilidades referentes a cada atributo selecionado no sistema em estudo. Assim, pode-se estabelecer o efeito relativo de cada atributo na utilidade total.

Este trabalho enfatiza a avaliação de softwares interativos tanto a partir de critérios ergonômicos e de qualidade amplamente divulgados e reconhecidos quanto a partir das preferências dos usuários, essas obtidas através das técnicas de preferência declarada. Isso porque, no processo de avaliação de softwares, seja para adoção ou para o desenvolvimento de novos, percebe-se a importância não apenas de questões de cunho técnico ou quantificáveis, mas também daquelas de cunho qualitativo, que despertam o interesse de quem efetivamente usa o produto com frequência.

## **1.2. Objetivo Geral**

O objetivo deste trabalho consiste no desenvolvimento de uma metodologia para avaliação ergonômica e de qualidade de software interativo baseada em técnicas de preferência declarada. A metodologia proposta é composta por dois módulos. No primeiro módulo é feita a avaliação de critérios básicos que todo software deve contemplar; no segundo são avaliadas as características específicas do software, sendo essas identificadas previamente a partir de experimentos de preferência declarada. Ao final da metodologia é



possível quantificar, em termos percentuais, qual o grau de conformidade do software avaliado segundo os critérios considerados.

Assim, o que se pretende com essa metodologia é que a mesma possa servir como auxiliar na tomada de decisão quanto à adoção de um software. Uma característica interessante da mesma é a alta flexibilização, já que as questões pertinentes à avaliação podem ser ponderadas de acordo com o grau de importância que cada um delas tem em cada situação específica. Além disso, ela pode servir também para concebedores de software avaliarem a performance de um protótipo, onde as respostas obtidas a partir da interação com futuros e potenciais usuários podem servir como *feedback* para a equipe de concepção e implementação do software.

### **1.3. Objetivos Específicos**

São objetivos específicos:

- Sistematizar recomendações ergonômicas e normas de qualidade para avaliação de software;
- Incorporar características comportamentais dos usuários em potencial na avaliação de um software específico;
- Propor uma métrica para o tratamento das informações obtidas;
- Identificar a utilidade máxima a ser alcançada por um software específico segundo critérios considerados relevantes;
- Quantificar a utilidade de um software específico que represente o grau de conformidade segundo os critérios considerados;
- Analisar uma área de estudos para aplicação da metodologia; no caso, a área escolhida foi a de roteirização de veículos;
- Levantar os atributos relevantes para softwares de roteirização de veículos;
- Aplicar efetivamente a metodologia em uma situação real, isto é, em um software de roteirização em específico; e
- Avaliar os resultados obtidos a partir da aplicação da metodologia proposta.

#### **1.4. Procedimentos Metodológicos**

Em todo trabalho de pesquisa inicialmente existe a necessidade de se definir as áreas de interesse para que possa ser realizada sobre elas uma ampla revisão bibliográfica, incluindo pesquisas em fontes fidedignas da *world wide web*. Nesse trabalho, as áreas de estudo são a ergonomia de software e a qualidade de software. Nestas duas áreas, a revisão bibliográfica foi feita com o intuito de identificar as metodologias existentes para avaliação de software e as lacunas existentes, tanto em nível de consistência teórica quanto de operacionalidade.

No desenvolvimento da metodologia de avaliação de software proposta utilizam-se as técnicas de preferência declarada para consideração da opinião dos usuários em potencial no processo de escolha de um software. Tais técnicas também tiveram que ser estudadas, visando identificar a operacionalização das mesmas.

Para mostrar a aplicabilidade da metodologia, é necessário obter dados comportamentais sobre os futuros usuários de softwares de roteirização. Para isso, é necessário um levantamento dos atributos relevantes para a área de análise escolhida, para só então ser possível efetivamente delinear o experimento, considerando o número de atributos, o número de níveis e a necessidade do uso de blocos para a realização efetiva da pesquisa de campo.

Como há um grande número de características consideradas importantes na avaliação de softwares roteirizadores, procede-se à segmentação do conjunto de atributos, idealizando-se um delineamento experimental dividido em 3 conjuntos. Essa segmentação segue a estratégia de Integração Hierárquica da Informação (*Hierarchical Information Integration*) sugerida por Louviere (LOUVIERE e TIMMERMANS, 1990) para tratar de casos como esse.

Como o objetivo da metodologia proposta é avaliar a qualidade propriamente dita de softwares, considerando para isso características operacionais, de funcionalidade e de interface, o atributo “custo” não é utilizado. Em geral os custos influenciam um processo de adoção de um software, especialmente se eles forem significativamente diferentes para produtos praticamente semelhantes. Ressalta-se, no entanto, que a metodologia tem um cunho mais amplo, podendo ser usada para proporcionar aos fabricantes um conjunto de características relevantes a serem incorporadas quando da concepção e desenvolvimento de

um software. Quando usada para decisão sobre adoção de um software dentre diversas ofertas disponíveis no mercado, pode-se proceder a avaliação de qualidade inicialmente para, a seguir, fazer-se a análise do fator preço. Ou, de forma contrária, pode-se elencar um conjunto de softwares cujo preço seja compatível com a capacidade de investimento desejada e, a seguir, fazer a análise da qualidade dos mesmos.

Uma vez obtidos os dados através da pesquisa de preferência declarada, deve-se estimar os coeficientes das funções utilidade. Para isso é utilizado o princípio estatístico clássico da verossimilhança. Tal atividade se dá a partir de um software já pronto, desenvolvido para uma tese de doutorado desta mesma instituição (SOUZA, 1999).

A avaliação dos resultados obtidos é feita através da análise de diversas medidas estatísticas, que medem a eficiência do processo e a validade dos dados obtidos.

A aplicação prática é feita avaliando-se um software existente no mercado que é bastante defendido e, segundo alguns usuários, alcança os objetivos almejados por softwares para esse determinado fim (no caso, a roteirização de veículos).

### **1.5. Estrutura do Trabalho**

O texto descrevendo a metodologia proposta neste trabalho, bem como a fundamentação teórica que embasa a mesma, é organizado em sete capítulos, sendo o primeiro este capítulo introdutório.

No Capítulo 2 são apresentados os fundamentos da Ergonomia de Software, dando ênfase a algumas das mais reconhecidas metodologias para concepção e avaliação de sistemas interativos, que são aquelas desenvolvidas a partir dos fundamentos da corrente francesa de estudos ergonômicos.

No Capítulo 3 são introduzidos os conceitos básicos sobre qualidade de software. Em seguida são apresentadas as normas mais importantes para avaliação da qualidade de produtos de software, identificando, em cada uma delas, quais características de qualidade devem ser consideradas relevantes e por isso devem ser avaliadas.

No Capítulo 4 são apresentadas as Técnicas de Preferência Declarada, essas de fundamental importância para o desenvolvimento da metodologia proposta.

A metodologia desenvolvida neste trabalho é apresentada no Capítulo 5. Detalham-se as atividades que foram realizadas para sua elaboração e apresenta-se, por fim, sua estrutura.

No Capítulo 6 apresenta-se a aplicação da metodologia proposta na avaliação de softwares para roteirização de veículos. São relatados todos os procedimentos seguidos bem como os resultados obtidos. Um exemplo de aplicação é também apresentado para ilustrar o seqüenciamento dos procedimentos de avaliação segundo o ferramental proposto.

No Capítulo 7 são apresentadas as conclusões do trabalho, elencando as contribuições do mesmo e sugerindo também caminhos possíveis a serem seguidos em pesquisas posteriores.

Finalmente, são apresentadas as referências utilizadas no desenvolvimento deste trabalho e citadas no mesmo, bem como os respectivos anexos e apêndices.

## **CAPÍTULO 2 – ERGONOMIA DE SOFTWARE**

### **2.1. Introdução**

A ergonomia é uma ciência que tem como principal objetivo conceber e avaliar produtos e ferramentas que possam ser utilizados com máximo conforto, segurança, satisfação e eficiência. Ela é considerada como uma ciência multidisciplinar, pois, para alcançar seu objetivo, busca integrar conhecimentos de áreas tais como: Psicologia, Fisiologia, Antropologia, Engenharia, Medicina e Segurança do Trabalho.

Essa ciência, também chamada de Fatores Humanos, teve início na Inglaterra no fim da década de quarenta, onde um grupo de cientistas e pesquisadores começou a perceber a importância de se considerar fatores humanos e do ambiente de trabalho para cada vez mais melhorar o desempenho, eficiência e segurança dos sistemas e produtos por eles desenvolvidos. Com o propósito de formalizar a existência dessa nova ciência, na segunda reunião do grupo, já em 1950, foi adotado o neologismo Ergonomia, derivado das palavras gregas *ergon* e *nomos*, que significam, respectivamente, trabalho e regras ou normas. Assim, é fundada a *Ergonomics Research Society*, sociedade ainda existente e que mantém seus alicerces na definição de ergonomia como a ciência que trata da interação da situação tecnológica e de trabalho do ser humano. Esta cita ainda que a ergonomia

é uma abordagem que põe as necessidades e capacidades humanas como foco do desenvolvimento de sistemas tecnológicos. Seu objetivo é assegurar que o ser humano e a tecnologia trabalhem em completa harmonia, com os equipamentos e tarefas alinhados às características humanas (*THE ERGONOMICS SOCIETY*, 2003 – tradução livre).

Em 1957 foi fundada nos Estados Unidos a *Human Factors Society of America*, após dois anos de encontros de profissionais da Associação de Engenharia Aeromédica de Los Angeles e da Sociedade de Engenharia Humana de San Diego. Com o propósito de promover a descoberta e a troca de conhecimentos relativos às características dos seres humanos que são aplicadas ao design de sistemas e produtos em geral, esta sociedade, que em 1992 teve seu nome trocado para *Human Factors and Ergonomics Society*, defende o uso sistemático destes

conhecimentos para alcançar compatibilidade entre pessoas, máquinas e ambiente no design de sistemas interativos eficientes, seguros e fáceis de usar (*HUMAN FACTORS AND ERGONOMICS SOCIETY*, 2003).

No Brasil os primeiros estudos sobre ergonomia foram feitos na década de sessenta na Universidade de São Paulo (USP), sendo Itiro Iida o autor da primeira tese brasileira sobre o assunto.

Iida (1997, p.1) define a ergonomia como sendo o estudo da adaptação do trabalho ao homem. O autor considera trabalho no sentido mais amplo da palavra, abrangendo como posto de trabalho, além das máquinas e equipamentos utilizados, “toda a situação em que ocorre o relacionamento entre o homem e seu trabalho”. Esta definição vem reforçar a idéia de que o trabalho é que precisa ser ajustado à pessoa, pois se sabe que em certos casos o homem tem que se adaptar a situações incômodas, o que pode gerar ineficiências, erros, situações de estresse e até mesmo danos físicos e mentais.

Assim, para que a concepção ou avaliação ergonômica de produtos ou sistemas seja realizada, é então necessário que o ergonomista (a pessoa responsável pelos fatores ergonômicos da concepção ou avaliação destes produtos ou sistemas) conheça o homem (operador) e seu trabalho (tarefa). Ele precisa estar a par de questões quanto à personalidade do operador, sua capacidade de comunicação, sua formação profissional, sua experiência com o trabalho e o ambiente e a frequência com que ele realiza certas tarefas. Quanto ao trabalho em si o ergonomista deve saber diferenciar a tarefa prevista para ser executada, da atividade efetivamente realizada na prática.

Neste sentido a Associação Brasileira de Ergonomia alerta que

é preciso que os ergonomistas tenham uma abordagem holística de todo o campo de ação da disciplina, tanto em seus aspectos físicos e cognitivos, como sociais, organizacionais, ambientais, etc. Frequentemente esses profissionais intervêm em setores particulares da economia ou em domínios de aplicação específicos. Esses últimos caracterizam-se por sua constante mutação, com a criação de novos domínios de aplicação ou do

aperfeiçoamento de outros mais antigos (ASSOCIAÇÃO BRASILEIRA DE ERGONOMIA, 2003).

A área de atuação da ergonomia é bastante ampla, com um vasto campo de aplicação que vai desde a ergonomia física à ergonomia cognitiva e organizacional. No presente estudo será enfocada mais especificamente a ergonomia de software, visando principalmente a concepção e avaliação de sistemas interativos.

## **2.2. Ergonomia de Software**

Para *The Ergonomics Society* (2003), uma das aplicações da ergonomia é o desenvolvimento de equipamentos e sistemas computacionais que sejam mais fáceis de usar e cuja probabilidade de ocorrência de erros na operação seja pequena. De maneira simplificada pode-se dizer então que a ergonomia de software concentra seus esforços nas condições de utilização de um software por seus usuários.

Com a crescente invasão das novas tecnologias no cotidiano do ser humano, cada vez mais indivíduos dependem da tecnologia informatizada. Hoje as pessoas deparam-se com computadores em casa, nas lojas, em ambientes escolares, de lazer e, claro, no ambiente de trabalho. Praticamente todas as pessoas que tiveram algum contato com a tecnologia computacional já passaram por alguma dificuldade de interação e já experimentaram falhas ou insucessos. Isso porque muitas vezes o designer do sistema não tem conhecimento suficiente sobre o usuário e a tarefa que este deseja realizar; muitas vezes a preocupação dele é simplesmente de apresentar um produto bonito, agradável aos olhos do usuário. É claro que a interface (meio que media a interação entre o homem e a máquina) precisa ser bonita e agradável, mas, sobretudo, ela deve ser funcional (ter utilidade) e de fácil uso (ter usabilidade).

Para Hix e Schulman (1991, p. 77) a utilidade indica o que a ferramenta pode fazer, qual o estilo da interface, as técnicas e características que podem ser reproduzidas pela ferramenta para alcançar os objetivos da aplicação da interface. Já a usabilidade indica quão bem a ferramenta executa suas possíveis funções em termos de facilidade de uso (“uma subjetiva mas quantitativa taxa de quão fácil ou difícil é a ferramenta”) e de performance

humana (“uma taxa objetiva de quão eficientemente a ferramenta pode ser usada para executar as tarefas”).

Em geral a expectativa dos usuários de informática é de encontrar, nos produtos que adquirem, um bom funcionamento aliado a uma boa usabilidade. O bom funcionamento pressupõe rapidez, precisão e acerto dos sistemas; usabilidade, por seu lado, pode ser melhor compreendida em termos de qualidades básicas como utilidade, intuitividade, facilidade de utilização e eficiência (GAINES e SHAW, 1987; CYBIS, 1994; GAMEZ, 1998).

Para que a interface possa de fato ser útil e usável, além de esteticamente agradável, o designer precisa ter em mente qual o tipo de usuário irá utilizá-lo e quais tarefas serão executadas a partir dele. Uma vez implementado, o sistema apresenta uma imagem física própria, percebida pelo usuário através das interações. Se a imagem do sistema não corresponder à estrutura mental de resolução do usuário, ele poderá ter problemas, sendo incapaz de prever as consequências de suas ações, além de que nenhum efeito de apresentação poderá maquiar as carências do sistema (COUTAZ, 1990 *apud* VALENTIN, VALLERY e LUCONGSANG, 1993, p. 19).

Apesar de cada ser humano ter características singulares, exigindo diferentes abordagens para a realização de tarefas, a população como um todo compartilha uma série de atributos mentais equivalentes. Portanto, antes de se preocupar com os requisitos individuais de cada usuário, o designer de um sistema precisa conhecer os aspectos fundamentais da cognição humana comuns a todos. Para Raskin (2004) é preciso

dominar a ergonomia da mente se quisermos desenvolver interfaces que provavelmente funcionem bem. Por incrível que pareça, frequentemente estamos cegos para nossos próprios limites mentais; devemos contar com a cuidadosa experimentação e observação para descobrir as fronteiras de nossas próprias habilidades mentais (tradução livre).

Sperandio (1983, p. 118) definiu a ergonomia de software como sendo concernente a todos os aspectos da interface homem computador (IHC) que dependem da programação, tendo relacionado como pontos a serem considerados com cautela:

- Os tempos de resposta do computador;
- Os códigos usados no diálogo;



- O vocabulário do diálogo;
- As regras sintáticas; e
- As formas de trocas de informação entre o computador e o usuário.

Para o caso específico de concepção de um sistema interativo, cita ainda a necessidade de atenção especial com os procedimentos para se efetuar a tarefa, a documentação e as instruções que explicam a maneira de usar o mesmo.

Pesquisas realizadas pelo governo de Québec no Canadá (*GOVERNEMENT DU QUÉBEC*, 1997) resultaram em um guia de concepção ergonômica de interação pessoa-sistema (IPS), voltado principalmente àqueles que concebem e desenvolvem protótipos de IPS nas áreas de saúde e serviços sociais. Este guia, dividido nas fases de Análise, Concepção e Avaliação, pretende determinar as funcionalidades de um sistema de informação e avaliar sua compatibilidade com o trabalho do usuário, considerando as necessidades deste, bem como as da organização na qual o sistema é adotado.

Barthet (1988), apoiada em conceitos da ergonomia e da psicologia cognitiva, propõe uma lista não exaustiva de recomendações ergonômicas para os parâmetros de interface de sistemas interativos, com especial atenção a questões:

- Do seqüenciamento das operações;
- Da linguagem de interação;
- Dos dispositivos de entrada e de apresentação;
- Do tempo de resposta;
- Do tratamento de erros; e
- Do tipo de ajuda.

Vicente (1999), baseado na visão de Rasmussen (1986; *apud* VICENTE, 1999), trabalha com o conceito de engenharia cognitiva com o intuito de analisar, desenvolver e avaliar sistemas baseados em computador. Segundo o autor, uma das principais características que distinguem a engenharia cognitiva de outras abordagens como a da Interface Homem Computador (HCI), da Ergonomia e da Psicologia, é a grande ênfase em estudos da semântica do domínio do trabalho. Assim, para que um sistema computacional seja efetivo (isto é, seguro, produtivo e saudável) ele deve integrar restrições ambientais e cognitivas. Isso quer

dizer que inicialmente faz-se necessário realizar uma análise explícita das restrições que o ambiente impõe sobre as ações para aí sim poder desenvolver uma interface compatível com o modelo mental, estratégias e preferências dos usuários.

O autor faz uma diferenciação entre dois tipos de abordagens para avaliação de um sistema computacional:

- Abordagem normativa: técnicas de avaliação que prescrevem como o sistema deve se comportar, fazendo isso a partir da análise da tarefa prescrita. Técnicas como estas são bastante usadas pela corrente americana de HCI.
- Abordagem descritiva: técnicas que descrevem como o sistema efetivamente se comporta na prática, sendo essas importantes para o entendimento das atividades que os usuários realmente executam e o que gostariam de poder fazer. Técnicas com esta são amplamente divulgadas e utilizadas há bastante tempo pela corrente francófona de ergonomia.

Porém, o autor ressalta que os sistemas computacionais não devem ser desenvolvidos ou avaliados somente baseados em prescrições como normas e nem somente visando as práticas nas quais os usuários estão engajados. Segundo ele, há necessidade de uma abordagem mais formativa para desenvolver e avaliar sistemas computacionais, abordagem essa que seja capaz de especificar os requisitos que devem ser satisfeitos para que o sistema se comporte da maneira desejada, segundo, claro, as restrições ambientais e o modelo mental dos indivíduos que farão uso do mesmo. Isso é o que Vicente (1999, p. 109) chama de criar condições para uma “adaptação produtiva”, com os usuários “terminando o design”. Concluindo, pode-se salientar, segundo as idéias do autor, que abordagens formativas focam na identificação dos requisitos – tecnológicos e organizacionais – que precisam ser satisfeitos se uma ferramenta for suportar um trabalho efetivamente. Embora esses requisitos não especifiquem um novo design, eles podem ser altamente informativos e bastante úteis, pois podem ser usados para desconsiderar outras alternativas de design ou mesmo outros softwares que tenham esse formato.

Estes e tantos outros autores (COUTAZ, 1990; VALENTIN, VALLERY e LUCONGSANG, 1993; NIELSEN, 1993; EKLUND, 1997; CYBIS *et al*, 1998; MOÇO, 1996; SCAPIN e BASTIEN, 1997; BURKHARDT, 1998; SHNEIDERMAN, 1998; PARASURAMAN, 2000) têm trabalhado no desenvolvimento de metodologias para conceber

ou avaliar sistemas interativos de qualidade, tarefa esta bastante complexa e árdua, haja vista que as percepções de utilidade e usabilidade variam bastante entre os indivíduos e de acordo com a situação de uso do sistema em questão. Mais além vão autores como Hendrick (2003), Worthy (2002, *apud* STANTON e BABER, 2003) e Beevis (2003) que atualmente realizam estudos sobre o custo da adoção de modelos ergonômicos no desenvolvimento de sistemas. Os autores pretendem mostrar, com pesquisas como estas, que os investimentos feitos na aplicação da ergonomia podem trazer benefícios que refletem e muito na eficiência, na produtividade e na redução de acidentes. Outro estudo interessante na área é o desenvolvido por Norton (2003) que levanta a questão da customização ergonômica dos ambientes computacionais, considerando os diferentes modelos mentais dos usuários, principalmente em se tratando de casos onde os usuários não tiveram grandes oportunidades de lidar com tecnologia.

A metodologia proposta no presente trabalho apresenta alguns fundamentos adotados pela corrente francófona de ergonomia. Em vista disso, os principais trabalhos de autores dessa corrente são apresentados de uma forma mais detalhada a seguir. Estas são baseadas nas experiências desenvolvidas por membros da comunidade de ergonomia e de interface homem-computador e no conhecimento disponibilizado em documentos como guias gerais de design, listas de verificação (*checklists*), padrões e heurísticas. Como os critérios adotados em cada uma dessas metodologias variam, de um autor para outro, e são altamente dependentes da interpretação de quem os aplica, pouco se fez no sentido de validá-los por completo. Mesmo assim, tais metodologias são bastante aceitas na comunidade de ergonomia e seria um erro aqui não considerar suas contribuições. Assim, essas metodologias, juntamente com pesquisas de preferência declarada, servirão como embasamento para o desenvolvimento de uma nova metodologia de avaliação de software interativo, o que vem a ser a proposta deste trabalho.

## **2.3. Metodologias para Concepção e Avaliação de Sistemas Interativos**

### **2.3.1. Critérios Ergonômicos de Scapin e Bastien**

Critérios ergonômicos podem ser entendidos como direções a seguir nas escolhas de um projeto, escolhas essas que podem garantir um comportamento mais eficiente da parte da interface com o usuário.

Scapin e Bastien (1997) trabalham com a idéia de um sistema interativo com qualidade ergonômica, um conceito mais amplo do que a definição comum de usabilidade. Para os autores o conceito de qualidade ergonômica de sistemas interativos cobre todos os aspectos do sistema que influenciam no cumprimento total da tarefa. Assim, ele cobre a usabilidade no sentido de facilidade de uso, isto é, quanto à facilidade com que o usuário pode alcançar suas metas de interação (que usualmente se referem aos modos, características da interface, diálogo etc). Também cobre a chamada utilidade, isto é, quanto à forma na qual o usuário pode alcançar suas metas (que usualmente se refere a aspectos funcionais de interação como funções, objetos, dados etc).

Na perspectiva da engenharia de software, pode-se dizer que qualidade ergonômica cobre não somente a apresentação clássica, controle do diálogo e aspectos de aplicação da interface, mas também alguns aspectos mais elementares: aqueles que influenciam o usuário no alcance de suas metas (SCAPIN e BASTIEN, 1997, p. 228).

Para avaliar então a qualidade ergonômica de um sistema interativo, os autores desenvolveram um conjunto chamado por eles de Critérios Ergonômicos. Este conjunto foi baseado em resultados experimentais disponíveis e em exame exaustivo em um longo conjunto de diretrizes individuais de diversos pesquisadores da área.

Inicialmente os dados pesquisados foram traduzidos em recomendações e regras de produção e posteriormente agrupados em conjuntos de acordo com sua pertinência. Após alguns ajustes foi estabelecido um conjunto constituído por oito critérios principais, alguns subdivididos, resultando ao final em dezoito critérios elementares, identificados neste texto por um asterisco (\*). O quadro 2.1 ilustra esta estrutura.

Cada critério foi formalmente definido, tendo também sua pertinência justificada. Além disso, para cada critério elementar, foram feitas recomendações, que nada mais são do que exemplos do que os autores sugerem que deve ser analisado no sistema.

1. Condução a) Presteza * b) Agrupamento e distinção de itens b.1) por localização * b.2) por formato * c) Feedback imediato * d) Legibilidade *
2. Carga de Trabalho a) Brevidade a.1) Concisão * a.2) Ações Mínimas * b) Densidade informacional *
3. Controle Explícito a) Ações explícitas do usuário * b) Controle do usuário *
4. Adaptabilidade a) Flexibilidade * b) Experiência do usuário *
5. Gestão de Erros a) Proteção contra erros * b) Qualidade das mensagens de erro * c) Correção de erros *
6. Consistência *
7. Significado dos Códigos *
8. Compatibilidade *

Quadro 2.1 – Critérios ergonômicos de Scapin e Bastien

### 2.3.1.1. Condução

A condução se refere aos meios disponíveis para avisar, orientar, informar, instruir e guiar o usuário na interação, visando facilitar o aprendizado, melhorar a performance e diminuir a ocorrência de erros. Estes meios em geral são mensagens, alarmes e rótulos que podem informar o usuário em qualquer momento onde ele se encontra na seqüência da interação ou no alcance de sua meta, quais são suas possíveis ações e as conseqüências destas,

além de outras informações adicionais possíveis. Este critério é dividido em 4 sub-critérios: presteza, agrupamento e distinção de itens, *feedback* imediato e legibilidade.

a) Presteza \*

A presteza é o convite à inicialização do sistema, caracterizado pelos meios disponíveis para guiar o usuário a realizar ações específicas, informando as formas de acesso ao sistema, as alternativas possíveis, o atual estado do sistema e as formas de ajuda. Uma boa presteza auxilia o usuário a navegar pelo sistema, evitando comandos indesejados e conseqüentemente erros.

b) Agrupamento e distinção de itens

É a forma de organização visual das informações, levando em conta a topologia e as características gráficas dos itens dispostos na tela, para que seja possível verificar semelhanças e diferenças entre eles. Uma boa organização dos itens na tela (ordenamento, posicionamento e distinção) garante que o usuário entenda melhor e possa memorizar mais facilmente as informações. Este sub-critério é dividido em dois critérios elementares: agrupamento e distinção por localização e por formato.

b.1) por localização \*

Trata especificamente do posicionamento relativo dos itens, devendo ficar claro quando estes pertencem ou não a uma mesma classe e também indicando as diferenças entre as classes. O ordenamento de textos, imagens e comandos respeitando alguma ordem lógica (alfabética, por frequência de uso ou relevância) facilitam o aprendizado e a memorização.

b.2) por formato \*

Trata especificamente do aspecto gráfico dos itens, tais como forma e cor. Da mesma forma que o critério anterior, o aspecto gráfico dos itens deve realçar semelhanças entre os itens e diferenças entre as classes de itens, para que o usuário possa entender melhor a relação entre eles e, conseqüentemente, possa ser melhor conduzido através do sistema.

c) Feedback imediato \*

O *feedback* do sistema se refere às respostas que este dá para as ações do usuário. Estas respostas devem prover informações referentes à operação realizada e seus consequentes resultados. Os tempos de resposta devem ser apropriados e consistentes de acordo com as diferentes ações tomadas e, dentro das possibilidades, o menor possível. Isso porque qualidade e rapidez são dois fatores que levam o usuário a confiar e se sentir satisfeito com o uso do sistema; caso contrário, um feedback atrasado ou mesmo a inexistência desse pode fazer com que o usuário suspeite de falhas operacionais.

d) Legibilidade \*

Diz respeito às características léxicas das informações apresentadas na tela que podem dificultar ou atrapalhar a leitura, como brilho, contraste entre a fonte e o fundo, tamanho da fonte, espaçamento, parágrafo, comprimento das linhas entre outros. Para decidir quanto à forma desses itens, o conhecimento das características cognitivas e de percepção do usuário é essencial, pois se sabe que, por exemplo, letras claras em fundo escuro e textos escritos em caixa alta são mais difíceis de serem lidos. É importante ainda ressaltar que não são consideradas nesta análise as mensagens de erro nem o *feedback*. As recomendações feitas pelos autores quanto a este critério de Condução são apresentadas na tabela 2.1.

#### 2.3.1.2. Carga de Trabalho

A carga de trabalho é o conteúdo do trabalho perceptivo e cognitivo resultante das tarefas a serem executadas. Diz respeito aos elementos da interface que podem reduzir esta carga e aumentar a eficiência do diálogo. Divide-se em dois sub-critérios: brevidade e densidade informacional.

Tabela 2.1 – Recomendações para a Condução

<b>CRITÉRIO ELEMENTAR</b>	<b>RECOMENDAÇÕES</b>
<b>Presteza</b>	Indicar o formato e os valores adequados para as entradas de dados (para datas disponibilizar o campo --/--/--)
	Exibir as unidades de medidas em que os dados devem estar
	Indicar todas as informações sobre o estado da interação
	Fornecer um rótulo para cada campo de dados
	Quando o campo é limitado indicar as dimensões deste
	Dispor um título para cada janela
	Fornecer orientação e ajuda <i>on line</i>
<b>Agrupamento e distinção de itens - por localização</b>	Organizar os itens em listas hierárquicas
	Organizar as opções de menu em função de sua aplicação
	Quando várias opções são apresentadas, organizar os itens de forma funcional ou relevante
<b>Agrupamento e distinção de itens - por formato</b>	Distinguir visualmente áreas que tenham diferentes funções
	Distinguir visualmente os campos de dados e seus rótulos
<b>Feedback imediato</b>	Mostrar na tela todas as entradas de dados do usuário (mesmo que use símbolos como *)
	Mostrar uma mensagem ao usuário de que o sistema voltou a seu estágio prévio sempre que ocorrer interrupção no processamento de dados
	Fornecer informações sobre o estado do processamento de informações quando este foi muito longo
<b>Legibilidade</b>	Centralizar os títulos
	Escrever os rótulos em letras maiúsculas
	Distinguir o cursor dos outros itens
	Quando o espaço de texto é limitado, dispor o texto em linhas longas (e não em várias linhas curtas)
	Exibir texto contínuo em colunas largas, com pelo menos 50 caracteres a linha
	Empregar a justificação à esquerda
	Manter as palavras “intactas”, com menos quebras ou hífen



a) Brevidade

Esse diz respeito à carga de trabalho, perceptiva e cognitiva, tanto para entradas e saídas individuais, quanto para um conjunto de ações necessárias para se alcançar um objetivo. Com este sub-critério deseja-se de alguma forma medir a carga de trabalho relacionada à leitura e ao número de passos de cada ação, tornando estes o mais sucinto possível. Essa necessidade de brevidade se dá porque a capacidade da memória de curto termo é limitada; assim, quanto menores forem as entradas menores serão as chances de ocorrência de erros. Além disso, com itens sucintos, o tempo de leitura é substancialmente menor. O critério brevidade está sub-dividido em dois critérios elementares, concisão e ações mínimas e não analisa aqui as mensagens de erro nem o *feedback* ao usuário.

a.1) Concisão \*

A concisão se dá pela brevidade da carga de trabalho perceptiva e cognitiva das entradas e saídas individuais. Quanto menores e mais concisas forem as entradas menor é a probabilidade de erros.

a.2) Ações Mínimas \*

Deseja-se com esse critério limitar o número de ações necessárias para se alcançar uma meta no software, evitando assim uma sobrecarga de trabalho, o que pode vir a gerar erros.

b) Densidade informacional \*

Esse critério se refere à carga de trabalho do usuário, isto é, como este percebe e processa as informações apresentadas como um todo e não individualmente. A densidade informacional é de extrema relevância, pois quando há excesso ou mesmo falta de informações, a performance do usuário é em geral prejudicada. Por isso, todos os itens necessários devem estar disponíveis e, pelos mesmos motivos citados acima, itens que não estejam relacionados com a tarefa devem ser removidos.

Na tabela 2.2 são apresentadas as recomendações para o critério Carga de Trabalho.

Tabela 2.2 – Recomendações para a Carga de Trabalho

<b>CRITÉRIO ELEMENTAR</b>	<b>RECOMENDAÇÕES</b>
<b>Concisão</b>	Permitir ao usuário entradas curtas de dados
	Quando os códigos têm mais de quatro caracteres, usar expressões mnemônicas ou abreviações
	Para dados numéricos a entrada de zeros à esquerda não deve ser necessária
	Quando há uma unidade de medida associada a um campo, incluir a mesma como parte do campo de dados
<b>Ações Mínimas</b>	Minimizar os passos para selecionar uma opção no menu
	Não requerer do usuário dados que podem ser obtidos do sistema
	Evitar entradas de dados que incluam pontuação
	Exibir, para entrada de dados, valores <i>default</i> apropriados
	Quando várias páginas estiverem envolvidas, possibilitar acesso direto a uma página específica
<b>Densidade Informacional</b>	Disponibilizar somente dados necessários e que serão usados
	Não utilizar dados que precisem de transformações de unidades
	Usar o mínimo de quantificadores na linguagem de consulta
	Não fazer com que o usuário tenha que memorizar dados exatos de uma tela para outra
	Prover processamento automático de dados para que o usuário não tenha que calcular e entrar com dados que possam ser derivados de dados já acessíveis ao sistema

### 2.3.1.3. Controle Explícito

Esse se refere ao mesmo tempo ao processamento das ações explícitas do usuário e ao controle que este tem sobre o processamento destas ações pelo sistema. Erros e ambigüidades são reduzidos quando o usuário define explicitamente suas entradas e estas ficam sob seu controle. Além disso, ele acaba por se sentir mais a vontade quando tem o controle sobre o diálogo com a máquina. O critério pode ser analisado então através das ações explícitas e do controle do usuário.

#### a) Ações explícitas do usuário \*

Aqui o foco está na relação entre o processamento do computador e as ações do usuário, sendo que o primeiro processa apenas as ações quando isto for requisitado. Desta

forma o usuário aprende e entende melhor o funcionamento do sistema e, com um sistema mais previsível, passa a cometer menos erros.

b) Controle do usuário \*

Esse critério salienta a importância do usuário ter o controle do processamento do sistema. Por isso, todas as possíveis ações do usuário devem ser previstas e as opções proporcionadas apropriadamente. Deve ser do usuário a opção de interromper, cancelar, pausar ou continuar um procedimento. Assim, o aprendizado é favorecido e os erros são minimizados.

Na tabela 2.3 estão elencadas as recomendações sugeridas para o critério em questão.

Tabela 2.3 – Recomendações para o Controle Explícito do Usuário

<b>CRITÉRIO ELEMENTAR</b>	<b>RECOMENDAÇÕES</b>
<b>Ações explícitas do usuário</b>	Requerer sempre que o usuário tecle ENTER para inicializar o processamento de dados
	Selecionar o menu por dupla ativação (primeira seleciona e segunda ativa a opção selecionada)
	Requerer sempre que o usuário tecle ENTER para efetivar as entradas de comandos
<b>Controle do usuário</b>	Permitir que o usuário entre com os dados em seu ritmo
	O cursor não deve se mover automaticamente sem o controle do usuário, exceto em casos óbvios como no preenchimento de formulários
	Permitir que o usuário controle as páginas que aparecem na tela
	Permitir que o usuário interrompa ou cancele um procedimento
	Disponibilizar a opção CANCELAR que tenha o efeito de apagar qualquer mudança feita, voltando à versão anterior

#### 2.3.1.4. Adaptabilidade

A adaptabilidade se refere à capacidade do sistema se comportar contextualmente de acordo com as necessidades e preferências do usuário. Quanto mais diversificados forem os meios de se realizar uma tarefa, maior a probabilidade de um usuário específico encontrar uma forma que lhe seja agradável e fácil. Sendo assim, diferentes procedimentos, opções e comandos devem estar disponíveis, para que a interface possa se adaptar ao homem. Este critério deve ser analisado sob dois sub focos: quanto à flexibilidade e quanto a consideração da experiência do usuário.

##### a) Flexibilidade \*

Aqui devem ser analisados os meios disponíveis para que o usuário modele a interface de acordo com seus hábitos, necessidades e estratégias de trabalho. Quanto mais diversificados forem os meios disponíveis para se realizar uma tarefa, maior será a probabilidade do usuário escolher e dominar um deles durante a aprendizagem.

##### b) Experiência do usuário \*

Um sistema deve poder se adaptar a diferentes usuários no que diz respeito a sua experiência com o mesmo, já que as informações que usuários novatos ou experientes precisam são em geral diferentes. Resumidamente, para aqueles sem experiência o software deve poder oferecer ações simples e passo-a-passo, enquanto que para os *experts* deve proporcionar teclas de atalho que agilizem o processamento. O ideal é que um software não considere apenas esses dois níveis e sim que tenha uma escala gradual para “medir” a experiência.

As recomendações quanto ao critério Adaptabilidade são apresentadas na tabela 2.4.

#### 2.3.1.5. Gestão de Erros

Esse critério diz respeito a todas as possíveis maneiras de evitar, alertar e corrigir erros como entrada de dados inválida ou com formato incorreto, comandos escritos em sintaxe errada ou ações indesejadas. Por ser um assunto dos mais delicados, toda atenção deve ser dada à prevenção de erros, à qualidade das mensagens de erro e à forma de correção de erros.

Tabela 2.4 – Recomendações para a Adaptabilidade

CRITÉRIO ELEMENTAR	RECOMENDAÇÕES
<b>Flexibilidade</b>	Quando possível, disponibilizar ao usuário meios de controlar a configuração das telas do sistema
	Quando valores <i>default</i> não foram definidos, permitir que o usuário os defina, remova ou mude
	Quando informações apresentadas forem desnecessárias, permitir que o usuário as descarte temporariamente
	Permitir que o usuário mude a seqüência das entradas de dados de acordo com suas preferências
	Quando o formato de um texto não puder ser previsto com antecedência, dispor meios de definir e salvar os formatos que o usuário venha a precisar
	Permitir que o usuário defina nomes para os campos de dados que ele venha a criar
<b>Experiência do usuário</b>	Prever atalhos para os usuários experientes
	Prever navegação passo-a-passo para os usuários inexperientes
	Disponibilizar diferentes tipos de diálogo para diferentes níveis de experiência dos usuários
	Disponibilizar diferentes tipos de diálogo para diferentes necessidades dos usuários
	Quando a condução é lenta para o usuário experiente, possibilitar que ele “pule” tais procedimentos
	Permitir que o usuário requisite mais informações sobre erros de acordo com seu nível de conhecimento

## a) Proteção contra erros \*

Refere-se as formas disponíveis para detectar e prevenir entrada de dados, comandos e ações incorretas, principalmente aquelas com conseqüências destrutivas. É importante, durante a concepção de um software, que sejam previstos todos os erros possíveis para que não ocorra, durante o processamento do sistema, uma falha grave ou uma pane no sistema.

## b) Qualidade das mensagens de erro \*

Uma mensagem de erro construtiva deve explicitar, de forma clara e bem escrita, a natureza do erro e as ações que devem ser tomadas para corrigí-lo. Sendo assim, uma boa mensagem de erro promoverá o aprendizado, ensinando como um certo erro pode ser prevenido ou resolvido.

## c) Correção de erros \*

Os erros causam menores danos quando são fácil e imediatamente corrigidos. Assim, as formas de correção dos erros devem ser simples e estar sempre claras para o usuário.

As recomendações dos autores quanto a este critério são apresentadas na tabela 2.5.

Tabela 2.5 – Recomendações para a Gestão de Erros

<b>CRITÉRIO ELEMENTAR</b>	<b>RECOMENDAÇÕES</b>
<b>Proteção contra erros</b>	Quando o usuário termina uma sessão e há alguma transação pendente ou dados serão perdidos, solicitar confirmação
	Proteger os rótulos de campos para evitar mudanças acidentais
	Proteger os campos que contêm informações para evitar mudanças acidentais
	Prever os possíveis erros, incluindo entradas de dados acidentais
<b>Qualidade das mensagens de erro</b>	Quando o usuário seleciona uma tecla de função inválida, apresentar uma mensagem indicando a função correta
	Adotar mensagens de erro orientadas à tarefa
	Dispor mensagens de erro o mais específicas possível
	Apresentar mensagens de erro breves, porém informativas
	Adotar um vocabulário neutro, não personalizado nem repreensivo ou humorístico
<b>Correção de erros</b>	Possibilitar a edição dos comandos durante a digitação
	Requerer do usuário a correção somente do comando ou da parte digitada equivocadamente
	Permitir que o usuário faça suas correções direta e imediatamente

### 2.3.1.6. Consistência \*

Este critério elementar, que também pode ser denominado de coerência ou mesmo homogeneidade, se refere ao modo como os códigos, nomes e formatos da interface são mantidos em contextos semelhantes e diferenciados em outros contextos. A falta de consistência entre uma tela e outra é uma causa forte de rejeição dos usuários, pois quando um sistema não respeita certos padrões, ele pode se tornar cansativo, demorado, ambíguo e até mesmo ilegível.

As recomendações dos autores para esse critério são apresentadas na tabela 2.6.

Tabela 2.6 – Recomendações para a Consistência

<b>CRITÉRIO ELEMENTAR</b>	<b>RECOMENDAÇÕES</b>
<b>Consistência</b>	Dispor os títulos no mesmo local
	Desenhar as tela em formatos similares
	Usar procedimentos similares para acessar as opções de menu
	Utilizar as mesmas pontuações e construções de frases
	Dispor as entradas de dados e comandos no mesmo local
	Padronizar os campos de entrada de dados

### 2.3.1.7. Significado dos Códigos \*

Este critério qualifica a relação entre conteúdo e o código que o referencia. Os códigos e denominações devem ter relação semântica com as ações que representam. A adoção deste critério se justifica pelo fato de que codificações significativas são mais acessadas e melhor memorizadas, enquanto que aquelas pouco significativas em geral levam a operações indesejadas.

As recomendações dos autores para esse critério são apresentadas na tabela 2.7.

Tabela 2.7 – Recomendações para o Significado dos Códigos

<b>CRITÉRIO ELEMENTAR</b>	<b>RECOMENDAÇÕES</b>
<b>Significado dos códigos</b>	Adotar títulos distintos e significativos
	Tornar regras de abreviações explícitas
	Usar códigos significativos e familiares ao usuário

### 2.3.1.8. Compatibilidade \*

O critério compatibilidade refere-se a relação entre as características do usuário (memória, percepção, hábitos, competências, idade, expectativas etc) e da tarefa que lhe foi designada. A eficiência é aumentada quando: os procedimentos adotados são compatíveis com a psicologia do usuário e respeitam suas expectativas, as traduções, transposições, interpretações ou referências a documentos são minimizadas e quando a informação é apresentada diretamente sob a forma que será utilizada. Compatibilidade diz respeito também ao grau de similaridade entre diferentes ambientes e aplicações.

Na tabela 2.8 são apresentadas as recomendações feitas pelos autores para esse critério.

Tabela 2.8 – Recomendações para a Compatibilidade

CRITÉRIO ELEMENTAR	RECOMENDAÇÕES
<b>Compatibilidade</b>	Assegurar que a organização das informações apresentadas em um documento está conforme a organização dos dados a entrar
	Compatibilizar o diálogo com a linguagem do usuário
	Formatar a entrada de datas de acordo com o costume do local
	Apresentar rótulos, <i>prompts</i> e mensagens de condução familiares e orientadas à tarefa
	Apresentar unidades de medida familiares ao usuário
	Apresentar dados textuais, mensagens e instruções em formato convencional de impressão

Os autores afirmam que, da forma como foi elaborado este conjunto de critérios ergonômicos, são garantidas sua validade e completude, pois o conjunto sintetiza a maioria das recomendações disponíveis em se tratando de interação homem-computador.

Este conjunto foi e ainda tem sido amplamente experimentado visando sua validação como método de análise ergonômica de interface homem-computador. (BASTIEN e SCAPIN, 1992), inclusive comparando-o com outros conjuntos (BASTIEN, SCAPIN e LEULIER, 1999). Sendo assim, os autores consideram que os critérios são válidos, confiáveis e úteis para, em particular, aumentar a performance de ergonomistas avaliadores (BASTIEN e SCAPIN, 1995).



Apesar desta forte tendência por adotar tais critérios, vale ressaltar que, de acordo com os próprios autores, esses não podem ser considerados de todo completos no caso de se ter tarefas muito específicas ou usuários altamente especializados. Nestes casos é necessária então a adoção de alguns critérios adicionais mais sofisticados que reflitam o contexto do usuário e da tarefa. Em outras palavras, a avaliação ergonômica de sistemas interativos pode frequentemente requerer outros métodos de avaliação; uma abordagem baseada em dimensões como critérios ergonômicos acaba por não substituir outros métodos de avaliação, mas sim constituir um complemento para tais métodos. (BASTIEN e SCAPIN, 1992).

### **2.3.2. Diagnóstico de Software Interativo Segundo Barthelet**

Em sua obra *Logiciels Interactifs et Ergonomie* Barthelet (1988) sugere algumas recomendações ergonômicas para os parâmetros da interface de sistemas interativos de uso profissional ou direcionados para o grande público. Para o desenvolvimento de tais recomendações a autora integra elementos provenientes da psicologia cognitiva e da ergonomia, haja vista a necessidade de se obter o maior número de dados sobre o tratamento das informações, tal como este é feito pelo usuário, e sobre a forma de diálogo deste com o computador.

A psicologia cognitiva é relevante por considerar a hipótese de que a resolução de uma tarefa é organizada segundo o modelo de planificação hierárquica (RAMOS, 1995; *ERGONOMICS*, 1998) . Além disso, considera que:

- Diferentes tipos de usuários, por razões variadas, não utilizam o software do mesmo modo;
- Há uma diferença significativa entre a lógica de funcionamento (como o software foi projetado para ser usado) e a lógica de utilização (como o software efetivamente é usado);
- Há necessidade de distinguir a tarefa prevista da tarefa efetivamente realizada pelo usuário;
- As características da memória de curto termo do ser humano têm relação direta com o seu diálogo com a máquina; e
- Uma parcela do aprendizado passa pela criação de automatismos.

Já a ergonomia tem relevância por observar:

- A seqüência de operações autorizada pelo software;
- A linguagem de interação homem-computador ao nível léxico e sintático;
- Os dispositivos de entrada de dados e comandos disponíveis;
- Os dispositivos de apresentação das informações e dos resultados;
- Os tempos de resposta percebidos pelo usuário;
- O tratamento de erros; e
- As ajudas disponíveis tanto para uso do software quanto para aprendizagem.

Cabe ressaltar que os conceitos da psicologia cognitiva e da ergonomia, apesar de terem sido considerados separadamente, não são de todo independentes, uma vez que seu objeto de estudo, a aplicação interativa, é o mesmo.

O modelo proposto pela autora para a concepção da interface de aplicações interativas visa integrar três pontos de vista, segundo o analista, o usuário e o programador. Neste modelo a visão do analista, chamada de Representação Conceitual, descreve a lógica geral do novo sistema, detalhando como ocorrerá o tratamento de informações e sua variabilidade. Esta visão integra os elementos da psicologia cognitiva diretamente ligados ao usuário. A visão do usuário, denominada Representação Externa, corresponde à maneira que o usuário vê e manipula o software. Esta representação pode ser dividida em duas partes: a primeira correspondendo aos elementos da psicologia cognitiva que têm relação com a interação homem-computador e a segunda compreendendo a definição de todos os elementos específicos da ergonomia da interface. Já a Representação Interna, ou o ponto de vista do programador, nada mais é do que a implementação das duas representações citadas anteriormente.

A relação entre tais representações e as considerações da psicologia cognitiva e da ergonomia é apresentada na figura 2.1.

Para cada um destes pontos relevantes quanto à ergonomia da interface levantados pela autora, foram detalhadas recomendações que são apresentadas de forma sintetizada a seguir.

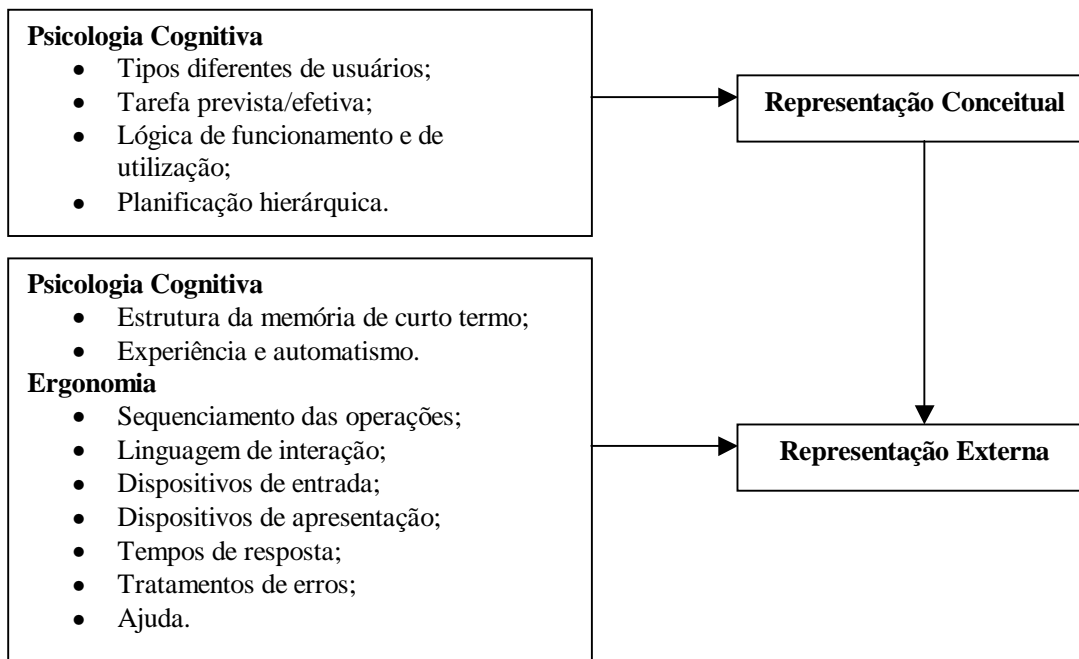


Figura 2.1 – Aportes da psicologia cognitiva e da ergonomia em relação às representações conceitual e externa (Barthet, 1988)

### 2.3.2.1. Recomendações ergonômicas para os parâmetros da interface

As recomendações ergonômicas para os sete parâmetros da interface elencados pela autora não são exaustivas, e como ela diz, combinações dessas podem levar a contradições parciais que só podem ser levantadas por uma experimentação junto aos usuários.

#### a) Seqüenciamento das operações

O problema principal está na adequação entre a ordem das operações fixada pela máquina e aquela necessária para o operador efetuar sua tarefa, independentemente do ambiente ou da experiência do usuário. Para tanto se faz necessário observar:

- O tipo de encadeamento (livre, guiado ou automático);
- As possibilidades do encadeamento variar segundo o grau de experiência e o perfil do usuário;
- A possibilidade de acesso a todas as informações a qualquer momento;
- A possibilidade de anular, interromper ou sair a qualquer momento;
- A possibilidade de salvar atividades já realizadas para futuramente retomá-las do mesmo ponto;
- A possibilidade de chamar uma operação qualquer a partir de outra e posteriormente voltar à primeira; e

- A possibilidade de transferir informações entre uma operação e outra.

Em nível de interface, essas características devem ser traduzidas em dispositivos de entrada e apresentação, vocabulário e sintaxe, mensagens de erros e ajudas, de forma a permitir que o usuário execute o mais facilmente possível o conjunto de possibilidades oferecidas.

#### b) Linguagem de interação

A linguagem de interação é a ferramenta que dá subsídios para o usuário exprimir, através de um vocabulário e uma sintaxe, as operações que ele deseja que a máquina efetue; além disso, é ela que permite que o usuário entenda as respostas fornecidas pela máquina.

Quanto ao vocabulário, é unânime entre os ergonomistas a preferência pelo emprego da linguagem dos especialistas no conteúdo ao qual o software se refere, em oposição à linguagem informática do programador. Porém, em alguns casos, como o da existência de ambigüidades entre a linguagem do especialista e do programador e o da criação de uma nova tarefa, há a necessidade de introduzir novos vocábulos. Nestes casos recomenda-se o uso de linguagem natural, podendo-se também utilizar ícones, desde que estes sejam representativos, não ambíguos e facilmente registráveis na memória. Especial cuidado deve ser tomado se forem adotados códigos numéricos e mnemônicos, devido à dificuldade de memorização e altos riscos de erros quando da manipulação dos mesmos.

Quanto à sintaxe, ou seja, às regras que exprimem os comandos, os dados ou a combinações destes, ela deve ser simplificada de forma a garantir simplicidade e homogeneidade, facilitando assim a memorização e evitando possíveis erros.

Dentre duas linguagens similares, a que será melhor lembrada e mais facilmente aprendida será aquela com menor número de regras. Na verdade, linguagens consistentes têm um pequeno número de regras gramaticais, enquanto, nas linguagens inconsistentes, as construções sintáticas têm pouco em comum entre si, não permitindo a sua redução para um pequeno número de regras (RAMOS, 1995, p. 149).

### c) Dispositivos de entrada

Os dispositivos de entrada nada mais são do que as formas que o usuário tem de enviar alguma informação para a máquina. Esta comunicação pode ser feita através de entrada de dados e comandos.

A entrada de dados em geral é feita através do teclado, mas há possibilidade também de se entrar com dados por comandos de voz ou através de toques na tela.

Os comandos podem ser dados a partir do teclado, usando palavras, teclas de atalho ou associação de teclas, esta última sendo mais eficiente quando o usuário é uma pessoa bastante habituada ao uso do teclado; podem ser dados também através de posicionamento do cursor e de toques na tela, maneira estas as mais fáceis de memorizar e as mais eficazes quando o usuário utiliza intensivamente o software.

### d) Dispositivos de apresentação

O trabalho interativo com uma máquina, mais especificamente com um computador, se dá principalmente frente a uma tela. Para conceber apresentações apropriadas para o usuário em questão, é preciso saber como o usuário explora o conjunto de telas a ele apresentado. Distinguem-se aqui dois casos:

- A pesquisa sistemática, onde o usuário explora toda a tela para achar o que procura. Neste caso as informações mais relevantes devem começar na coluna superior esquerda ou se situar na zona central da tela.
- A pesquisa seletiva, onde o usuário, já familiarizado com a tela, se fixa nas regiões desta onde ele sabe que irá encontrar a informação desejada. Neste caso as informações de mesma natureza devem estar agrupadas em zonas (de menus, de mensagens, de entrada de dados, de botões, etc) que aparecerão sempre padronizadas e na mesma localização.

Quanto à apresentação de dados, Barthet sugere que esta seja feita ou através de perguntas a serem respondidas ou através do preenchimento de um formulário. A primeira situação é mais utilizada em casos onde o programa é totalmente guiado pela máquina; já a segunda, mais flexível, permite que o usuário entre com os dados na ordem que lhe for conveniente.

Os comandos em geral devem ser apresentados na forma de um menu que pode estar sempre aparente ou aparecer somente quando solicitado, pode ser fixo ou dinâmico, estando disponíveis somente os comandos possíveis de serem utilizados naquele momento. Casos em que os comandos são dados através de uma linguagem própria (linguagem de comando) só devem ser implementados quando os usuários são altamente especializados e utilizam o software freqüentemente.

e) Tempos de resposta

O problema principal aqui é identificar qual o tempo de resposta ideal. É evidente que tempos de resposta muito longos necessitam ser acompanhados de mensagens para que o usuário possa eventualmente realizar outra tarefa enquanto aguarda. Além disso, tempos de resposta muito variáveis devem ser evitados para que o usuário não confunda pequenos atrasos com ineficiências de operação ou mesmo pense que ele realizou algum procedimento incorretamente. Assim, tanto a autora quanto outros especialistas no assunto concordam com os seguintes tempos:

- Até dois (2) segundos é o tempo de resposta ideal;
- De dois (2) a quatro (4) segundos há uma impressão de espera, mas causa pouco incômodo à memória de curto prazo do usuário; e
- Mais de quatro (4) segundos é um tempo muito longo se o diálogo necessita de memorização de curto prazo. Se o caso for este, deve-se fixar uma mensagem de espera na tela.

f) Tratamento de erros

Barthet distingue dois tipos de erros: os de execução, que se dão pela digitação errada, e os de intenção, que se dão pela má interpretação de comandos e procedimentos. Os erros, independentes do tipo, devem ser identificados o mais rapidamente possível por causa da volatilidade da memória de curto termo. Os erros de execução são facilmente detectáveis e corrigíveis; já os de intenção só são detectados mais tarde e sua correção depende da aprendizagem do usuário.

As mensagens de erro devem explicitar, sempre que possível, a causa do erro. Elas devem ser apresentadas sempre da mesma forma visando a homogeneidade, principalmente nos casos de usuários freqüentes e que utilizam a pesquisa seletiva sobre a tela.

Para a correção dos erros o usuário deve poder voltar, com auxílio, aonde se encontra o erro, podendo anular todo ou em parte o trabalho feito posteriormente. É importante dizer que a correção de erros é uma das tarefas mais complicadas quando se trata de sistemas informatizados, pois deve ser prevista quando da definição da seqüência de operações.

#### g) Ajuda

A autora cita Senach (*apud* BARTHET, 1988, p. 92) para identificar dois tipos de ajuda:

- Ajuda funcional por descrição de comandos, onde se descreve para cada comando sua definição e os efeitos da seleção do mesmo e
- Ajuda de utilização por reconhecimento dos planos de ação do usuário, onde se indica o conjunto de operações a percorrer em função do objetivo que o mesmo deseja alcançar.

A primeira é mais apropriada a usuários com experiência que precisem lembrar questões sintáticas ou semânticas de algum comando; a última é mais útil a usuários novatos que precisem ser guiados para a realização de uma tarefa.

#### 2.3.2.2. Diagnóstico de um Software Interativo

Barthet desenvolveu o método DIANE de diagnóstico de software interativo, indicado para ser usado por pessoas que não estejam diretamente envolvidos com a concepção do software, como um ergonomista, um consultor de informática ou mesmo por um futuro usuário. Para proceder o diagnóstico em geral dispõe-se dos dossiês de programação e análise e do software em si. Os dossiês de análise podem indicar como está estruturada a base de dados, a arborescência do menu, o encadeamento das operações e, eventualmente, a estrutura e design das telas. O dossiê de programação pode detalhar as entradas e saídas de dados, bem como a ocorrência de erros e como estes foram controlados. Por serem as informações encontradas um tanto superficiais, elas devem ser testadas com o uso do próprio software. Sendo assim, muitas vezes acaba sendo mais eficaz o diagnóstico feito direta e unicamente a partir do exame do software. A autora então sugere que este diagnóstico seja feito em duas etapas, sendo que em cada uma delas todas as telas do software e as ações tomadas devem ser transcritas, para que a seqüência de operações possa ser acompanhada, o que garante ao final que o software foi testado em sua totalidade. Na primeira etapa realiza-se o teste do software a

partir dos parâmetros do quadro de análise (tabela 2.9); na segunda etapa são feitos testes com usuários.

Tabela 2.9 – Síntese do quadro de análise de um software (Barthet, 1988)

<b>Parâmetros</b> (para cada tela)	<b>Elementos de teste</b>	<b>Teste com o usuário</b>
<b>Vocabulário</b>	Ambigüidade Regras de abreviação	Memorização Erros
<b>Sintaxe</b>	Homogeneidade Simplicidade	Erros Rapidez de execução Memorização
<b>Dispositivos de entrada</b>	Homogeneidade em relação aos outros materiais utilizados	Erros
<b>Dispositivos de apresentação</b>	Legibilidade da tela Homogeneidade da apresentação Pertinência das zonas	Lentidão na execução Fadiga
<b>Tempos de resposta</b>	Atrasos Mensagem de espera	Incômodo
<b>Tratamento de erros</b>	Erros detectados Legibilidade das mensagens Possibilidade de correção de erros	Confiabilidade no tratamento Diminuição do trabalho Desvio dos procedimentos
<b>Ajuda</b>	Tipo de ajuda Acesso a todo o momento Coerência do software	Facilidade de aprendizagem Facilidade de uso

O quadro de análise (*grille d'analyse*) baseia-se nas recomendações ergonômicas da interface vistas anteriormente e deve ser utilizado para uma análise bastante detalhada que deve ser feita em cada tela do sistema.

Quanto ao vocabulário é necessário se perguntar:

- Ele é ambíguo?
- É difícil de memorizar algo necessário?
- As abreviações, se existentes, são mnemônicas?

Quanto à sintaxe deve-se:



- Testar ações idênticas do usuário (estas devem ter o mesmo efeito em todo o programa);
- Testar homogeneidade do comando (um comando deve sempre realizar a mesma ação); e
- Verificar a simplicidade da mesma.

Poucos erros em geral são encontrados quando se trata dos dispositivos de entrada. Pelo fato de que as entradas de dados recebem especial atenção dos programadores, estas em geral são padronizadas e pouco variantes, facilitando assim a inserção de dados ou comandos.

Quanto aos dispositivos de apresentação fazem-se necessárias as questões:

- As informações na tela são legíveis?
- As informações de mesma natureza aparecem sempre no mesmo local e seguem o mesmo padrão em todas as telas?
- Há zonas distintas para objetos informacionais distintos?
- Se existem essas zonas distintas, elas são pertinentes para o usuário?

Quanto à ajuda, deve-se identificar os diferentes tipos de ajuda que o software oferece para o usuário e, para cada uma, verificar:

- Se há a possibilidade de o usuário ser guiado em todo o uso do software; e
- Se há coerência nas explicações da ajuda (as explicações sobre as ações dos comandos correspondem à realidade).

Algumas considerações ainda devem ser levantadas junto aos usuários quanto ao encadeamento entre as telas:

- Na arborescência do menu existe uma lógica de utilização?
- O encadeamento de telas é automático? Isso facilita ou não a interação?
- É possível interromper as operações? Em caso negativo, pode-se navegar facilmente pelas operações antes de efetivá-las?
- Comandos como Sair e Ajuda estão sempre disponíveis e produzem sempre o mesmo efeito?
- Caso algum comando esteja indisponível em algum momento, o usuário tem como identificá-lo como tal?

- Há homogeneidade de sintaxe, dispositivos de entrada e de apresentação?
- Há homogeneidade entre este software e outros utilizados em conjunto, no mesmo local de trabalho?

Feita a avaliação do software pelo ergonomista, as sugestões de melhora devem ser traduzidas em especificações operacionais para serem passadas aos informatas responsáveis pelo desenvolvimento do sistema. Espera-se que com estas informações, obtidas principalmente a partir dos diferentes pontos de vista de usuários que fariam uso do software, seja possível implementar um software mais homogêneo, totalmente adaptado à situação de trabalho e ao usuário.

### **2.3.3. Método Iterativo de Concepção de Valentin, Vallery e Lucongsang**

Estes três autores, especialistas em ergonomia de software industrial e empresarial, defendem a idéia de que a concepção de um software, para ser coerente, deve ser duplamente adaptada, tanto aos objetivos que a empresa almeja quanto às características da população usuária futura (VALENTIN, VALLERY e LUCONGSANG, 1993, p. 7). Assim como outros autores já citados, para a concepção de um sistema interativo, argumentam que é necessário conhecer o usuário e a tarefa a ser realizada através do software.

Os fundamentos teóricos desse método seguem os ideais da corrente francesa de ergonomia. Esta corrente centra seus princípios na análise da atividade dos operadores, sugerindo que esta deve ser feita sobre a situação real do trabalho ou, eventualmente, a partir de simulações as mais realistas possíveis.

Com isso, pode-se dizer que para a utilização do método iterativo de concepção, é desejável que o ergonomista conheça o local onde será utilizado o software, presencie como o operador procede, verifique as ações por ele realizadas, analise as informações por ele utilizadas e verifique sob que condições a tarefa foi efetuada. Além disso, o ergonomista pode ainda “provocar” o operador a verbalizar questões de seu interesse e estes comentários podem, futuramente, ajudar a compreensão dos objetivos do trabalho e dos raciocínios operacionalizados pelo usuário e suas causas.

Assim, para a operacionalização do método é necessário se dispor:

- Do software pronto, em protótipo ou maquete;
- Da documentação correspondente, quando esta existe;
- De ao menos uma estação de trabalho equipada que represente a situação a ser estudada;
- De um dos concebedores; e
- De usuários experientes nas tarefas realizadas pelo software.

Estando estes disponíveis, pode-se inicializar o método, que segue o esquema da figura 2.2.

As cinco etapas do método – reconhecimento do software, definição da população e dos cenários, coleta de dados, análise dos dados coletados e síntese dos resultados, são descritas na seqüência.

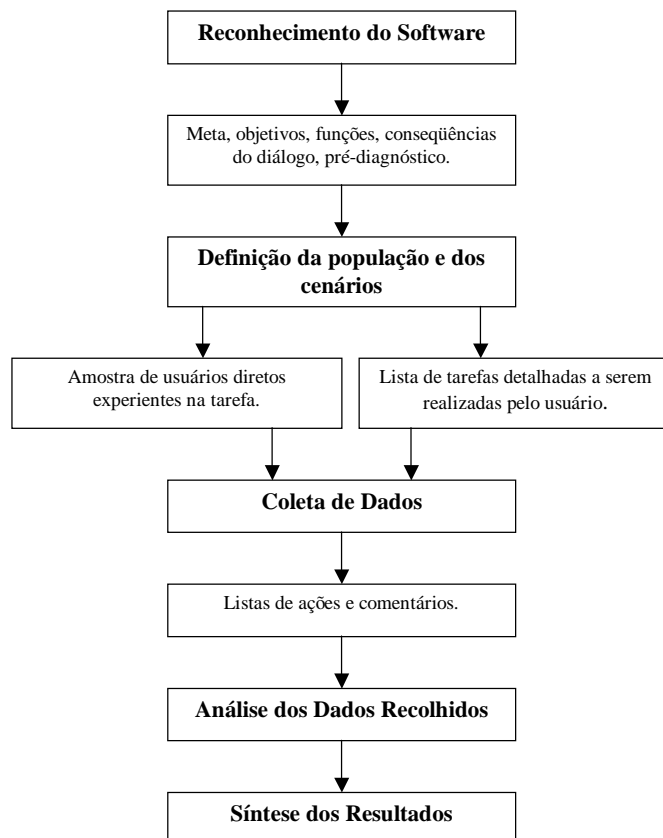


Figura 2.2 – Esquema geral do método (Valentin *et. al.*, 1993)

### 2.3.3.1. Reconhecimento do software

Nesta fase o ergonomista pretende tomar conhecimento do produto, realizar um pré-diagnóstico sobre a interface do diálogo e destacar os pontos que devem ser estudados mais detalhadamente com os usuários. Esse reconhecimento deve ser feito através de algumas ações como:

#### a) Entrevista com o concebedor

Permite que o ergonomista conheça: as tarefas almejadas pelo software, seu público alvo, as especificidades do software em relação aos produtos concorrentes, como são consideradas as necessidades do usuário, a assistência a ser oferecida ao usuário, os efeitos e as condições de trabalho na organização e o ambiente técnico necessário.

#### b) Demonstração do software

Se possível, após a entrevista com o concebedor, o ergonomista assiste a uma demonstração do software obtendo assim informações suficientes para posteriormente poder testar o produto de maneira mais autônoma. Com esta demonstração ele pode visualizar o tipo de diálogo, os menus, a organização geral das informações e as principais funcionalidades do software.

#### c) Testes

O ergonomista percorre as diferentes funções do software fazendo uma avaliação à priori das características ergonômicas da interface, baseado em seus conhecimentos de ergonomia cognitiva e interface homem-computador. A medida que percorre o software ele tenta responder questões quanto a apresentação das informações, os modos de captura, as funções perigosas, a gestão de erros, a documentação e a flexibilidade. Em termos gerais deve verificar se o software:

- Respeita os objetivos e conhecimentos do usuário;
- Limita a carga de memorização;
- Respeita a homogeneidade das apresentações e ações;
- Informa e guia, informando ao usuário o que ele fez e o que ele pode fazer;
- Respeita o vocabulário do usuário;
- Protege os comandos e funções perigosas;
- Fornece tempos de resposta curtos e estáveis;

- Dá possibilidades de interrupção e retomada de tarefas; e
- Constrói um diálogo adaptável ao perfil do usuário.

Uma lista mais detalhada de questões a serem respondidas pelo ergonomista nesta fase, sugerida pelos autores, é apresentada no Anexo 1.

d) Entrevista com os usuários

Neste primeiro contato com os usuários, participantes da equipe concebedora ou usuários com experiência na tarefa, o ergonomista pretende descobrir como o software irá influenciar na evolução e organização do trabalho.

Com os dados coletados nesta primeira etapa de reconhecimento, o ergonomista tem em mãos um pré-diagnóstico onde se identificam os objetivos do produto e suas principais funções. Assim, é feita uma primeira avaliação do software, centrada no diálogo onde se analisa, por exemplo, a homogeneidade, coerência e segurança do programa. Com isso, tem-se uma lista de pontos a serem esclarecidos com o usuário quando este fizer uso do software.

#### 2.3.3.2. Definição da população e dos cenários

Nesta etapa, com o auxílio do concebedor do software e de usuários experientes na tarefa, o ergonomista almeja definir o público alvo, constituindo uma amostragem representativa dos usuários futuros, e determinar as condições de utilização do software, criando roteiros de tarefas que melhor simulem a real situação de trabalho com o software. Ele deve definir então a população e os cenários.

a) População

De acordo com as informações obtidas na etapa de reconhecimento, o ergonomista escolhe, para efetuar as tarefas em situação de simulação, pessoas que estão ou efetivamente estarão nos postos de trabalho e que ao mesmo tempo sejam experientes na realização da tarefa; porém estas podem ter experiência ou não com o uso do software.

Os iniciantes no uso do software dão mais indicações quanto à facilidade de aprendizagem e simplicidade de utilização, já os usuários experientes fornecem mais elementos sobre a organização das funções e a repartição das informações.

O número de usuários utilizados na amostra é bastante relativo: este deve ser suficiente para que considere os diferentes perfis dos usuários e que possa diferenciar as observações gerais, porém este número não pode ser grande demais, pois isto acabaria por gerar uma enorme carga de trabalho e tempos muito longos. Os três exemplos de aplicação do método apresentados pelos autores foram feitos utilizando amostras de seis a doze pessoas.

Quanto à formação da amostra os autores alertam que qualquer que seja a amostra tomada, convém formalizar o modo de seleção e sua descrição. Isto permite recolocar no seu contexto os resultados obtidos e melhor definir o nível de generalização.

#### b) Cenários

Segundo os autores, um cenário é uma série de instruções correspondentes aos objetivos do trabalho a ser realizado com o software. Para a construção destes cenários o ergonomista precisa identificar quais as tarefas mais importantes de acordo com sua frequência ou incidência sobre a atividade a ser realizada. Esta identificação será feita de acordo com informações obtidas em quatro fontes:

- Nas hipóteses feitas pelo ergonomista no pré-diagnóstico, citando as vantagens e dificuldades ligadas à utilização do software;
- Nos objetivos que o concebedor havia dado, tomando cuidado para que os cenários fiquem estruturados não para o funcionamento do software e sim para a tarefa a ser realizada;
- Nos elementos sobre o trabalho real com o software, através de observação direta ou entrevistas, dependendo do tempo disponível para testes com usuários, do estágio em que se encontra o software e do nível de detalhamento desejado; e
- Em um exame do conjunto de funcionalidades do software, onde um usuário experiente na tarefa, para cada elemento da lista de funcionalidades catalogada com o concebedor durante o reconhecimento do software, atribui uma pontuação seguindo três critérios: frequência de uso, incidência sobre o resto da atividade e ganho estimado em relação ao mesmo trabalho efetuado sem o software.

A partir das informações obtidas dessas quatro fontes o ergonomista formula o cenário evidenciando os objetivos do usuário no planejamento de sua ação, sua lógica na execução da tarefa, seus procedimentos para o alcance dos objetivos e as condições necessárias para a aplicação destes procedimentos.

#### 2.3.3.3. Coleta de dados

Nesta etapa são reunidas as informações sobre a utilização do software, a partir da observação individual das atividades realizadas pela amostra da população, que segue os cenários criados na etapa anterior. Ele deve se atentar para:

a) A preparação da avaliação

Antes da avaliação propriamente dita sugere-se que seja feito um ensaio, com pelo menos um usuário, objetivando precisar a duração da avaliação e melhor planejá-la quando da análise do software por toda a amostra populacional.

b) A coordenação dos observadores

Quando forem utilizados vários observadores, há a necessidade de se realizar uma reunião de trabalho para, de alguma forma, padronizar as coletas. Os observadores, em geral ergonomistas e informatas, devem respeitar algumas regras, tais como:

- Não dar informações sobre o produto testado nem fazer comentários pessoais;
- Levantar as diferentes ações do usuário, solicitar explicações e anotar comentários correspondentes;
- Anotar a que correspondem as ações em relação ao software, tal como título da tela, função ou mensagem associada;
- Deixar o usuário seguir o caminho que desejar, mesmo que não o correto ou usual, anotando todas as ações e conseqüências;
- Não interferir quando o usuário não encontrar imediatamente a solução ou não souber como prosseguir, apenas chamar uma pessoa da equipe de concepção, anotar as explicações fornecidas e as ações que se seguiram;
- Não interpretar a causa das dificuldades do usuário, simplesmente anotar detalhadamente o procedimento do mesmo com o software.

c) O desenrolar da coleta

É indicado que seja feita inicialmente uma apresentação aos usuários que irão participar da simulação, falando da estrutura do software e seus objetivos, bem como do desenrolar da avaliação e dos resultados esperados.

Durante a avaliação cada observador acompanha um usuário, dando-lhe a instrução geral de um cenário e observando sua realização. São verificadas as ações do usuário e suas conseqüências, as funções utilizadas e as dificuldades encontradas. Os comentários feitos paralelamente ajudam a compreender o que guia o usuário, suas reações e hesitações.

Ao final da avaliação o observador submete ao usuário um questionário mais geral e subjetivo, que permite cobrir as funções que não foram levantadas no software. No Anexo 2 apresenta-se um exemplo de questionário sugerido pelos autores.

#### 2.3.3.4. Análise dos dados coletados

As etapas anteriores, desde o pré-diagnóstico do ergonômico e do levantamento das características do software com o concebedor, até as observações diretas e questionários aplicados com os usuários, resultam em diversas informações que o ergonômico deve processar e comparar com as recomendações ergonômicas, as exigências da situação de trabalho e as exigências técnicas da ferramenta computacional em si.

#### 2.3.3.5. Síntese dos resultados

Com a união das informações obtidas até então o ergonômico procede a síntese dos resultados, tendo por fim um documento onde constem:

- A identificação da demanda (nome do autor, tipo de demanda e objetivos de avaliação);
- A descrição do produto (ambiente técnico, contexto de uso e descrição funcional);
- As condições de avaliação (materiais e técnicas, população, contexto da avaliação);
- A definição de cenários de tarefas (descrição de cada cenário e motivação para a escolha destes);



- O roteiro de observações (descrição sumária do desenrolar da observação, descrição das estratégias dos usuários e síntese dos principais incidentes, suas causas e frequência);
- Os resultados de diferentes questionários (tabela recapitulativa);
- As recomendações e propostas para os modos de diálogo (funções, vocabulário e mensagens), a apresentação das informações (densidade e estrutura), as funcionalidades, o encadeamento das telas, a documentação, a ajuda, a formação necessária do usuário e a ligação com a organização do trabalho; e
- As conclusões.

Para encerrar, os autores enfatizam que a validade deste método de concepção de software está fortemente relacionada com o realismo dos cenários e a representatividade da população. Além disso, sugerem que, para a concepção de softwares bem adaptados ao modo de trabalho do usuário, este método de avaliação seja realizado pelo menos em dois momentos:

- No processo de concepção, a partir de uma maquete ou protótipo do software e
- Em um momento onde este, já bem desenvolvido, possa ser manipulado.

Através da análise dos métodos apresentados pode-se verificar a importância e complexidade envolvida nos processos de avaliação ergonômica de software. As metodologias aqui apresentadas seguem basicamente os conceitos da corrente francesa de ergonomia, portanto, têm uma abordagem mais descritiva (Vicente, 1999). Para se realizar então uma avaliação mais robusta de sistemas computacionais, faz-se necessário também conhecer as normas de qualidade de software, sendo as mais relevantes delas apresentadas no capítulo seguinte.

## **CAPÍTULO 3 – QUALIDADE DE SOFTWARE E AS NORMAS ISO/NBR**

### **3.1. Introdução**

Com a globalização e a abertura de mercado, a competitividade está cada vez mais acirrada. Para se destacar frente a seus concorrentes e aumentar progressivamente sua fatia de mercado, as empresas estão dando especial atenção aos anseios e às opiniões de seus clientes. Estes, cada vez mais conscientes de seus direitos de consumidores, buscam total satisfação na escolha de um produto ou serviço. Essa mudança no paradigma das empresas, focando o julgamento do cliente, fundamenta o conceito que hoje se entende por qualidade. De fato

qualidade e produtividade são metas cada vez mais perseguidas por organizações produtivas, quaisquer que sejam seus portes e setores em uma economia. Constituem-se conjugadamente, por assim dizer, na chave da competitividade inteligente, conducente aos maiores lucros e à máxima segurança no mercado (GOMES e OLIVEIRA, 1993, p. 1).

A qualidade de um produto ou serviço pode ser definida como a adequação deste ao uso pelo consumidor. Paladini (1997, p. 17) comenta que

ao enfatizar a “adequação ao uso”, nota-se que serão responsáveis pela qualidade todos os elementos que tiveram alguma participação, direta ou indireta, na produção do bem ou do serviço. Esses elementos são as pessoas, equipamentos, métodos, informações, ambientes etc. Envolvem o processo produtivo, em todas suas fases, desde o fornecimento de matéria-prima até a colocação do produto acabado na casa do cliente; a assistência técnica que se prestará a ele; as informações que serão repassadas aos clientes para permitir o uso correto do produto etc.

Essa adequação ao uso pode ser encarada também como satisfação das necessidades dos usuários, necessidades estas que precisam estar bem especificadas e podem variar com o tempo. Inclui-se nesta adequação ao uso aspectos de usabilidade, satisfação, disponibilidade, confiabilidade, manutenibilidade, economia e meio ambiente.

Na mesma linha de pensamento Weber (1999, p.1) se refere à qualidade e à produtividade como uma filosofia de gestão, que tem como objetivo conduzir a organização à melhoria de seus processos, visando obter por fim “produtos e serviços com desempenho, preço e disponibilidade adequados, totalmente orientados para as aspirações do cliente”.

Uma vez delimitados um processo, seus produtos e clientes, torna-se necessário identificar, junto aos últimos, quais as características de qualidade mais importantes de cada produto, ou seja, aquelas que agregam maior valor à sua satisfação. De fato

a verdadeira qualidade sempre se inicia ao saber o que os clientes querem e, possivelmente, aquilo que precisam ou esperam. A principal responsabilidade do cliente é decidir sobre os requisitos de avaliação, pois estes oferecem o padrão para medição da qualidade. Para tanto, é necessário que se faça uma grande interação com o usuário, uma vez que ele será o verdadeiro árbitro na avaliação da qualidade do produto (CAMPOS, PASSOS e VILLELA, 2002).

Para que a qualidade de um produto disponível ao consumo seja mais facilmente identificada pelo cliente é necessário que exista alguma classificação ou certificação oficial atestando a qualidade do bem ou serviço. Essas devem estar baseadas em padrões ou normas amplamente difundidas e reconhecidas; por exemplo, a classificação de estrelas em hotéis ou os selos SIF e ABIC para carnes e cafés respectivamente. Órgãos como a ISO (*International Organization for Standardization*), o INMETRO (Instituto Nacional de Metrologia, Normalização e Qualidade Industrial) e a ABNT (Associação Brasileira de Normas Técnicas) são órgãos normalizadores largamente conhecidos que, em suas áreas específicas, analisam e atestam a qualidade de produtos e serviços com o intuito de promover a competição entre concorrentes, incentivar inovações e, principalmente, proteger o consumidor.

No Brasil, em 1990, foi criado o Programa Brasileiro de Qualidade e Produtividade (PBQP), um movimento aberto à participação de toda a sociedade e que tem como objetivo alcançar a melhoria da qualidade de vida da população e a competitividade de bens e serviços produzidos no país. Atualmente a coordenação desse programa é feita pelo INMETRO e pelo Movimento Brasil Competitivo (MBC) e tem como eixo condutor as Metas Mobilizadoras Nacionais, formuladas pelo Governo em parceria com entidades representativas da sociedade. Essas metas refletem os anseios da população por qualidade em diferentes setores como educação, saúde, trabalho, ciência, tecnologia etc.

A qualidade é uma meta a ser perseguida por todas as organizações, públicas ou privadas, produtoras de bens ou serviços. No presente estudo será dada ênfase à questão da qualidade na informática, mais especificamente nos produtos de software.

### **3.2. Qualidade de Software**

Desde meados dos anos sessenta, a proliferação dos produtos de software (programas desenvolvidos para serem amplamente vendidos a clientes) e o posterior advento dos microcomputadores, fizeram com que o software se tornasse um componente decisivo para muitos produtos. Em decorrência desse fato passou-se a ter uma crescente e constante preocupação com a forma de se produzir um software. Isto porque um indivíduo, quando adquire um programa de computador, deseja que o mesmo opere sem defeitos, seja seguro, financeiramente acessível, não seja computacionalmente pesado e, claro, atenda às suas necessidades. Resumidamente ele deseja adquirir um programa de qualidade.

Segundo Pressman (1995, p. 724) a qualidade de software é a

conformidade a requisitos funcionais e de desempenho explicitamente declarados, a padrões de desenvolvimento claramente documentados e a características implícitas que são esperadas de todo software e/ou sistema profissional desenvolvido.

Rezende (1999, p. 97) completa dizendo que o software deve estar adequado à empresa, ao cliente e/ou usuário e efetivamente terá qualidade se, quando pronto, gerar informações com qualidade, isto é, adequadas, úteis, precisas, confiáveis, claras e oportunas.

Resumidamente, pode-se dizer que um software de qualidade é aquele que tem (ARTHUR, 1994 *apud* REZENDE, 1999, p. 96):

- Defeito zero, já que em aplicações ou sistemas críticos as conseqüências de uma falha podem ser desastrosas;
- Requisitos funcionais adequados à satisfação do usuário;
- Codificação estruturada e elegante, onde qualquer engenheiro de software possa efetuar modificações sem dificuldades;
- Desempenho satisfatório, onde o tempo de resposta seja adequado;
- Custo adequado, onde o investimento seja justo e propicie resultados e retorno;
- Desenvolvimento rápido e produtivo, onde equipe de desenvolvimento e usuário tenham elaborado um planejamento adequado e satisfatório; e
- Facilidade para o usuário, onde o mesmo possa manipular o software de forma acessível.

Essa visão de qualidade enfatiza muito mais as características técnicas do programa, como a lógica de funcionamento, do que a forma de utilização mais satisfatória para o usuário.

Sob o ponto de vista da Engenharia de Software, a qualidade de software é definida como

um conceito multi-dimensional que não é fácil definir de maneira simples (...) a parte crítica do planejamento da qualidade é a seleção dos atributos críticos de qualidade e o planejamento de como esses podem ser atingidos (SOMMERVILLE, 1997, P. 612).

Essa abordagem tem total relevância por se preocupar com as métricas da produtividade e qualidade de um software (DALE e VAN DER ZEE, 1992). Estas nada mais são do que “medidas do resultado do desenvolvimento do software como uma função do esforço aplicado e medidas de ‘adequação ao uso’ do resultado que é produzido”. (PRESSMAN, 1995, p. 60). Apesar da relevância, no presente trabalho não será dada ênfase à noção de qualidade segundo a Engenharia de Software.

Mais especificamente é possível definir qualidade de um produto de software como sendo “a totalidade de características de um produto de software que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas” (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - NBR 13596). Segundo Barreto (2000) as necessidades explícitas são as próprias condições e objetos propostos pelo produtor e as necessidades implícitas incluem as diferenças entre os usuários, a evolução no tempo, as implicações éticas, as questões de segurança e outras visões subjetivas.

Nos últimos anos, órgãos normalizadores nacionais e internacionais vêm estabelecendo normas para desenvolvedores de software que identificam as características de qualidade que um software deve contemplar. A ISO tem desenvolvido diversas normas para análise da qualidade do produto, do pacote e do processo de desenvolvimento do software. A ISO em conjunto com a IEC (*International Electrotechnical Commission*) e o SEI (*Software Engineering Institute da Carnegie Mellon University*) desenvolveram respectivamente o SPICE (*Software Process Improvement and Capability dEtermination*) e o CMM (*Capability*

*Maturity Model*) que são métodos para avaliação de processo de desenvolvimento de software muito bem aceitos pelos especialistas da área.

Existem normas específicas para a avaliação da qualidade dos produtos de software. Estas, em geral, além de considerarem a qualidade dos aspectos técnicos, consideram também as questões de qualidade da interface com o usuário e da qualidade ergonômica dos componentes do software. As principais normas referentes à avaliação da qualidade de produtos de software serão abordadas a seguir.

### **3.3. Normas ISO e NBR para Avaliação da Qualidade de Produtos de Software**

Produtos de software necessitam ser avaliados para que seja possível decidir se certas características de qualidade relevantes são satisfeitas pelo mesmo. Para avaliar a qualidade de um produto faz-se necessária a adoção de um conjunto de características de qualidade que descreva o produto e sirva de base para a avaliação.

Atualmente muitas instituições se preocupam em criar normas que possibilitem uma efetiva avaliação da qualidade de software, seja no que se refere aos processos de desenvolvimento destes ou a produtos já prontos e disponibilizados à aquisição e uso.

No presente trabalho serão enfatizadas as séries de normas NBR 13596 e NBR ISO/IEC 14598 que tratam especificamente da avaliação de produtos de software. Além disso, serão também feitas considerações relevantes sobre a norma NBR ISO/IEC 12119, referente a pacotes de software, e a norma internacional ISO/IEC 9241, referente às exigências ergonômicas para o trabalho com monitores de vídeo.

#### **3.3.1. A Norma NBR 13596**

Esta norma, intitulada Tecnologia da informação – Avaliação de produto de software – Características de qualidade e diretrizes para seu uso, foi baseada na norma internacional ISO/IEC 9126/1991, tornando-se válida a partir de maio do ano de 1996. Coerente com a definição de qualidade da NBR ISO 8402 (1996), intitulada “Gestão da qualidade e garantia da qualidade – Terminologia”, a norma é dirigida às pessoas envolvidas com aquisição, desenvolvimento, uso, suporte, manutenção ou auditoria de software.

A norma tem caráter geral e é altamente conceitual, o que faz com que a mesma seja aplicável em ambientes e condições variadas. Por ser uma pura tradução da ISO/IEC 9126, ela corrobora com a idéia de uniformização dos conceitos de qualidade, fazendo o uso de textos normativos comuns a todos os países.

Esta norma elenca um conjunto de características que devem ser verificadas em um software para que ele possa ser considerado de qualidade. São seis características gerais (Funcionalidade, Confiabilidade, Usabilidade, Eficiência, Manutenibilidade e Portabilidade) que por sua vez são desdobradas em sub-características. Os requisitos iniciais da equipe de elaboração da norma para a obtenção das características foram os seguintes:

- O conjunto de características em sua totalidade deveria cobrir todos os aspectos de qualidade de um software segundo as definições de qualidade da ISO;
- Não poderia existir a sobreposição de conceitos;
- O conjunto deveria ser formado contendo de seis a oito características, por questões de clareza e facilidade de manuseio; e
- Dever-se-ia considerar a terminologia já conhecida no meio de estudo.

A seguir são transcritas da norma as definições de cada uma das seis características e apresentadas, para cada uma delas, as sub-características associadas sugeridas no texto da norma.

#### 3.3.1.1. Funcionalidade

Conjunto de atributos que evidenciam a existência de um conjunto de funções e suas propriedades especificadas. As funções são as que satisfazem as necessidades explícitas ou implícitas. As sub-características associadas são:

- a) Adequação: atributos do software que evidenciam a presença de um conjunto de funções e sua apropriação para as tarefas especificadas.
- b) Acurácia: atributos do software que evidenciam a geração de resultados ou efeitos corretos ou conforme acordados.

- c) Interoperabilidade: atributos do software que evidenciam sua capacidade de interagir com sistemas especificados.
- d) Conformidade: atributos do software que fazem com que ele esteja de acordo com as normas, convenções ou regulamentações previstas em leis e descrições similares, relacionadas à aplicação.
- e) Segurança de acesso: atributos do software que evidenciam sua capacidade de evitar o acesso não autorizado, acidental ou deliberado, a programas e dados.

#### 3.3.1.2. Confiabilidade

Conjunto de atributos que evidenciam a capacidade do software de manter seu nível de desempenho sob condições estabelecidas durante um período de tempo pré-determinado. Suas sub-características são:

- a) Maturidade: atributos do software que evidenciam a frequência de falhas por defeitos no software.
- b) Tolerância a falhas: atributos do software que evidenciam sua capacidade de manter um nível de desempenho especificado nos casos de falhas no software ou de violação nas interfaces especificadas.
- c) Recuperabilidade: atributos do software que evidenciam sua capacidade de estabelecer seu nível de desempenho e recuperar os dados diretamente afetados, em caso de falha, e no tempo e esforço necessários para tal.

#### 3.3.1.3. Usabilidade

Conjunto de atributos que evidenciam o esforço necessário para se poder utilizar o software, bem como o julgamento individual desse uso, por um conjunto explícito ou implícito de usuários. As sub-características associadas são:

- a) Inteligibilidade: atributos do software que evidenciam o esforço do usuário para reconhecer o conceito lógico e sua aplicabilidade.



- b) Apreensibilidade: atributos do software que evidenciam o esforço do usuário para aprender sua aplicação (por exemplo: controle de operação, entradas, saídas).
- c) Operacionalidade: atributos do software que evidenciam o esforço do usuário para sua operação e controle da sua operação.

#### 3.3.1.4. Eficiência

Conjunto de atributos que evidenciam o relacionamento entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas. As sub-características desta são:

- a) Comportamento em relação ao tempo: atributos do software que evidenciam seu tempo de resposta, tempo de processamento e velocidade na execução de suas funções.
- b) Comportamento em relação aos recursos: atributos do software que evidenciam a quantidade de recursos usados e a duração de seu uso na execução de suas funções.

#### 3.3.1.5. Manutenibilidade

Conjunto de atributos que evidenciam o esforço necessário para fazer modificações especificadas no software. Como sub-características tem-se:

- a) Analisabilidade: atributos do software que evidenciam o esforço necessário para diagnosticar deficiências ou causas de falhas, ou para identificar partes a serem modificadas.
- b) Modificabilidade: atributos do software que evidenciam o esforço necessário para modificá-lo, remover seus defeitos ou adaptá-lo a mudanças ambientais.
- c) Estabilidade: atributos do software que evidenciam o risco de efeitos inesperados, ocasionados por modificações.
- d) Testabilidade: atributos do software que evidenciam o esforço necessário para validar o software modificado.

### 3.3.1.6. Portabilidade

Conjunto de atributos que evidenciam a capacidade do software de ser transferido de um ambiente para outro. Suas sub-características são:

- a) Adaptabilidade: atributos do software que evidenciam sua capacidade de ser adaptado a ambientes diferentes especificados, sem a necessidade de aplicação de outras ações ou meios além daqueles fornecidos para essa finalidade pelo software considerado.
- b) Capacidade para ser instalado: atributos do software que evidenciam o esforço necessário para sua instalação em um ambiente especificado.
- c) Conformidade: atributos do software que o tornam consonante com padrões ou convenções relacionadas à portabilidade.
- d) Capacidade para substituir: atributos do software que evidenciam sua capacidade e esforço necessário para substituir um outro software, no ambiente estabelecido para este último.

Barreto (2000) resumiu de maneira simples através de questões-chave o que significa cada uma dessas características e sub-características descritas acima, como se pode ver na tabela 3.1.

A norma NBR 13596 não define como dar nota a um software em cada um dos itens; para isso devem ser criados níveis de pontuação em cada uma das sub-características. Algumas características como o tempo de execução, o número de erros encontrados ou o tempo médio entre as falhas podem ser quantificados através de medições; o problema está naquelas características de caráter altamente subjetivo onde não há nenhuma forma de medição direta. Nestes casos há a necessidade de se criar métricas de software, ou seja, um “método e uma escala quantitativa que podem ser usados para determinar o valor que uma particularidade (*feature*) recebe em um produto de software específico” (NBR 13596, 1996, p. 2). De forma simplificada, por exemplo, podem ser feitas perguntas a serem respondidas com Sim, Praticamente Sim, Praticamente Não, Não e Não se Aplica, ou ainda podem ser feitas afirmações que ganharão pontuações que vão de zero a dez, dependendo do nível de desempenho alcançado pelo sistema.

Tabela 3.1 – Características e sub-características da NBR 13596 (Barreto, 2000)

<b>Característica</b>	<b>Sub-característica</b>	<b>Questão-chave para a sub-característica</b>
<b>Funcionalidade</b> (Satisfaz às necessidades?)	Adequação	Propõe-se fazer o que é apropriado?
	Acurácia	Faz o que foi proposto de forma correta?
	Interoperabilidade	Interage com os sistemas especificados?
	Conformidade	Está de acordo com as normas, leis, etc?
	Segurança de Acesso	Evita acesso não autorizado aos dados?
<b>Confiabilidade</b> (É imune a falhas?)	Maturidade	Com que frequência apresenta falhas?
	Tolerância à falhas	Ocorrendo falhas, como ele reage?
	Recuperabilidade	É capaz de recuperar dados em caso de falha?
<b>Usabilidade</b> (É fácil de usar?)	Inteligibilidade	É fácil de entender o conceito e a aplicação?
	Apreensibilidade	É fácil aprender a usar?
	Operacionalidade	É fácil operar e controlar?
<b>Eficiência</b> (É rápido e enxuto?)	Tempo	Qual tempo de reposta, a velocidade de execução?
	Recursos	Quanto recurso usa? Durante quanto tempo?
<b>Manutenibilidade</b> (É fácil de modificar?)	Analisabilidade	É fácil encontrar uma falha, quando ocorre?
	Modificabilidade	É fácil modificar e adaptar?
	Estabilidade	Há grande risco quando se faz alterações?
	Testabilidade	É fácil testar quando se faz alterações?
<b>Portabilidade</b> (É fácil de usar em outro ambiente?)	Adaptabilidade	É fácil adaptar a outros ambientes?
	Capacidade para ser instalado	É fácil instalar em outros ambientes?
	Conformidade	Está de acordo com padrões de portabilidade?
	Capacidade para substituir	É fácil usar para substituir outro?

É válido ressaltar também que cada uma das diferentes características pode ter pesos diferentes, já que a importância de cada característica de qualidade varia dependendo do tipo de software; por exemplo, em softwares interativos, a característica usabilidade tem maior importância.

Atualmente a norma NBR 13596 está sendo revisada para acompanhar as modificações feitas no seu documento de referência, a norma ISO/IEC 9126. Apesar de as características básicas citadas anteriormente não terem sido modificadas na sua essência, algumas pequenas alterações foram feitas:

- A inclusão da sub-característica Conformidade em todas as seis características;
- A inclusão da sub-característica Atratividade na característica Usabilidade; e
- A inclusão da sub-característica Capacidade de Coexistir na característica Portabilidade.

Além disso, foram incluídos documentos adicionais que descrevem as métricas externas (aquelas relativas ao uso do produto), as métricas internas (aquelas relativas a arquitetura do produto) e as métricas para qualidade de uso. Estas últimas, na verdade, representam a inovação da norma, já que através delas pretende-se verificar “a capacidade do produto de software de permitir a usuários específicos atingir metas especificadas com efetividade, produtividade, segurança e satisfação” (SCALET, 2001, p. 44).

Tanto em seu projeto original quanto no melhorado, o modelo de qualidade desta norma privilegia a visão do usuário do produto de software que, em geral, atua a partir da operação do sistema no qual o produto de software faz parte.

O processo de avaliação da qualidade de um software foi descrito na norma de forma sucinta. Posteriormente, para completar as instruções da norma, foi desenvolvida a norma ISO/IEC 14598 que permite uma avaliação padronizada das características de qualidade de um produto de software. Esta detalha como deve ser o processo de avaliação e em que este se baseia. Esta norma, com algumas partes já traduzidas para o português e adotada sob o título de NBR ISO/IEC 14598 – Tecnologia de informação – Avaliação de produto de software, será apresentada a seguir.

### **3.3.2. A Norma NBR ISO/IEC 14598**

Considerando que a classe de produtos de software é bastante diversificada, a norma ISO/IEC 14598, de 1998, fornece uma estrutura para avaliar a qualidade de qualquer tipo de produto de software, obtendo por fim “resultados quantitativos sobre a qualidade do produto de software que sejam compreensíveis, aceitáveis e confiáveis por quaisquer das partes interessadas”. (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - NBR ISO/IEC 14598-5, 2001, p. 2).

Esta norma apresenta-se dividida em seis partes:

- 14598-1: Parte 1: Visão Geral
- 14598-2: Parte 2: Planejamento e Gestão
- 14598-3: Parte 3: Processo para Desenvolvedores
- 14598-4: Parte 4: Processo para Adquirentes
- 14598-5: Parte 5: Processo para Avaliadores
- 14598-6: Parte 6: Documentação de Módulos para Avaliação.

Como dito anteriormente nesse texto, a tradução desta para a norma brasileira correspondente, a NBR ISO/IEC 14598, não foi totalmente concluída e divulgada; porém, as partes 1 e 5 acima citadas, mais relevantes para o presente estudo, foram validadas no final de setembro do ano de 2001.

De acordo com a norma, o processo de avaliação deve respeitar as seguintes características:

- Repetibilidade, onde avaliações repetidas do mesmo produto e sob as mesmas condições, pelo mesmo avaliador, produzam resultados que podem ser aceitos como sendo idênticos;
- Reprodutibilidade, onde avaliações do mesmo produto e sob as mesmas condições, por avaliadores diferentes, produzam resultados que podem ser aceitos como sendo idênticos;
- Imparcialidade, garantindo que não haja tendência a algum resultado particular;  
e
- Objetividade, garantindo que os resultados das avaliações sejam baseados em fatos concretos e não em crenças e opiniões do avaliador.

A norma, por descrever com detalhes os passos para que se avalie um produto de software, interessa:

- A desenvolvedores que pretendem melhorar a qualidade do seu próprio produto;
- A empresas que adquirem software por encomenda que pretendem avaliar a qualidade do produto antes da aquisição propriamente dita; e

- A empresas que fazem certificação de produtos, já que a norma facilitará a emissão de documento oficial, com fundamentação de qualidade de software aceita mundialmente.

Para efetivamente se avaliar a qualidade de um pacote de software, inicialmente devem ser estabelecidos os requisitos de avaliação, baseados nas características de qualidade definidas na norma ISO/IEC 9126, ou, no presente caso, na sua correspondente brasileira NBR 13596. Posteriormente deve-se especificar, projetar e executar a avaliação propriamente dita, seguindo o esquema apresentado pela norma, como pode ser visto na figura 3.1.

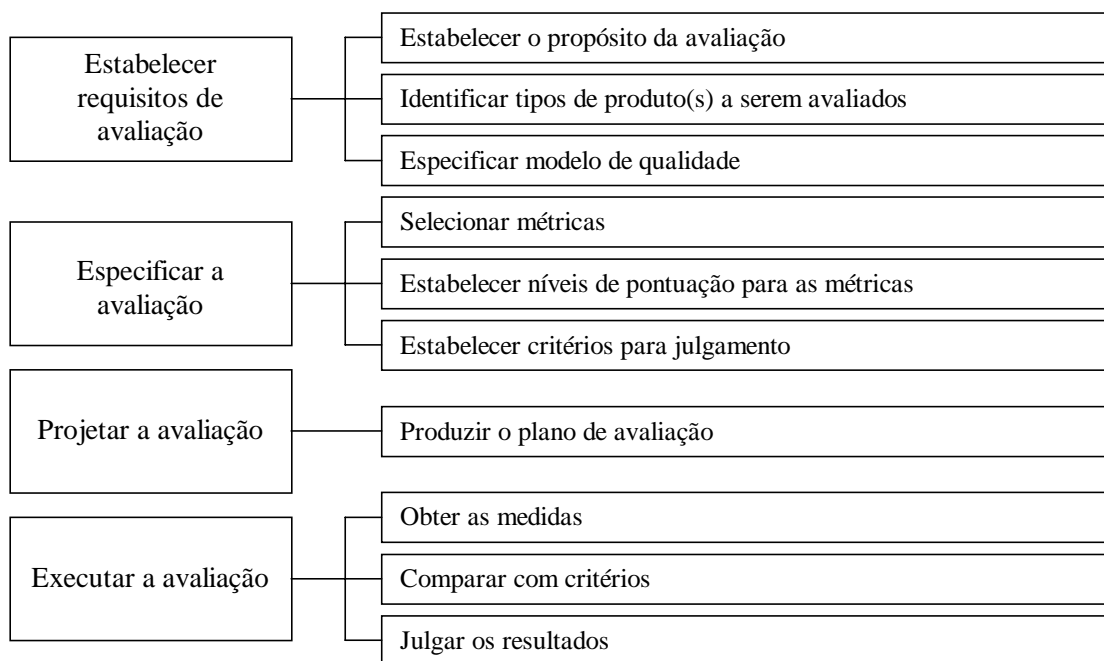


Figura 3.1 – Processo de Avaliação de acordo com a norma ISO/IEC 14598-1 (2001)

Cada uma das etapas apresentadas na figura 3.1 é descrita sucintamente a seguir.

### 3.3.2.1. Estabelecer requisitos de avaliação

Visando garantir que o produto em julgamento tenha a qualidade pretendida, isto é, que atenda as necessidades do usuário, deve-se descrever os objetivos de avaliação. Para tanto, é necessário pré-estabelecer 3 condições para a avaliação, já que é imprescindível de antemão saber qual o propósito da avaliação, quais os tipos de produto a serem avaliados e qual será o modelo de qualidade a ser seguido.

No estabelecimento do propósito da avaliação, deve ficar claro o motivo pelo qual está sendo feita a análise do produto. Com isto:

- Se for para a aquisição de um produto de software por encomenda, que esta seja feita respeitando as características de qualidade desejadas;
- Se for para o desenvolvimento de um novo produto, que este seja baseado nos requisitos de qualidade previamente estudados e acordados entre todos os membros da equipe responsável pelo desenvolvimento do software; e
- Se for para o fornecimento de produto de software, que este seja feito com a garantia de qualidade do mesmo quando comparado a outros produtos semelhantes.

Além disso, pode-se pretender obter resultados para serem usados pela equipe de organização que opera o sistema de software, para que seja possível, a partir desses, validar se os requisitos de qualidade são atendidos sob diferentes condições. Também, os resultados da análise podem servir de subsídio para que a equipe de manutenção do sistema onde o software opera possa validar a manutenibilidade e a portabilidade do mesmo.

O segundo requisito que deve ser identificado antes do início da avaliação propriamente dita é qual o tipo de produto a ser avaliado. Isso depende diretamente do propósito de avaliação previamente estabelecido e do estágio do ciclo de vida do produto, ou seja, se ele está em fase de desenvolvimento ou se já se encontra em sua forma final. Além disso, pode ser também interessante conhecer o hardware, o software e, se for o caso, o ambiente de rede no qual o produto será usado.

Para concluir esta primeira fase do processo de avaliação descrito na norma, é preciso ter especificado o modelo de qualidade que servirá de referência para a definição dos requisitos de qualidade do produto de software. Em geral este modelo é exatamente aquele encontrado na NBR 13596 com as seis amplas categorias de características de qualidade de software: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Como visto anteriormente, estas são divididas em sub-características que definem, em sua totalidade, a qualidade em uso.

Assim sendo, a priorização de um requisito de qualidade em detrimento de outros menos relevantes, neste estágio inicial do processo de avaliação, baseia-se essencialmente nas necessidades dos usuários (final, operador e mantenedor) e nos propósitos de avaliação.

### 3.3.2.2. Especificar a avaliação

Nesta fase o objetivo é definir a abrangência da avaliação e das medições a serem realizadas sobre o produto e seus componentes. Ela é dividida em três atividades: a seleção das métricas, o estabelecimento dos níveis de pontuação para as métricas e o estabelecimento de critérios para julgamento.

As métricas a serem selecionadas para avaliação dependem diretamente das características de qualidade esperadas de um produto e dos objetivos da avaliação. Se, por exemplo, a intenção com a avaliação for de identificar problemas que possam ser corrigidos, listas de verificação (*checklists*) e questionamentos a especialistas podem ser úteis para monitorar e controlar alterações de melhorias no software. Já se o objetivo for certificar a qualidade do produto, a comparação com outros produtos similares deve seguir um modelo de qualidade preciso, além de contar com medições, escalas e níveis de pontuação altamente rigorosos para cada métrica.

De acordo com o texto da norma, todo atributo do software, seja interno ou externo, desde que quantificável e correlacionado com uma característica, pode ser definido como uma métrica. Scalet (2001, p. 48) resume os tipos de métricas que podem ser:

- Internas: aquelas associadas à arquitetura do produto de software e que permitem prever a qualidade do produto final;
- Externas: aquelas mensuráveis quando o produto está em operação; e
- De qualidade em uso: aquelas que avaliam o efeito do uso do produto de software.

É importante que as métricas internas tenham relação com algum critério externo desejado, isto é, que elas realmente representem algo mensurável sobre a qualidade do software quando este está em uso.

A atividade seguinte é estabelecer um nível de pontuação, de forma a permitir que, baseado nele, o software seja classificado de acordo com as necessidades pré-estabelecidas. Assim, o software será considerado satisfatório em sua qualidade se estiver dentro de uma faixa de aceitação. Como sugestão, pode-se dividir a escala de pontuação em quatro categorias que representem: um produto inaceitável, um produto com as características mínimas aceitáveis, um produto aceitável e um produto que supera as expectativas. Estando na primeira categoria o produto é considerado insatisfatório, enquanto que estando nas três



últimas, o mesmo é considerado satisfatório, em maior ou menor grau, em termos de qualidade.

Por fim, para que seja possível julgar a qualidade, é preciso sintetizar os resultados medidos e pontuados. Para tanto deve-se ter um procedimento padrão com critérios bem especificados para cada uma das características e sub-características diferentes, além de considerações adicionais sobre tempo e custos, relacionados ao produto em um ambiente particular.

#### 3.3.2.3. Projetar a avaliação

Nesta etapa produz-se o plano de avaliação que descreva os recursos necessários para realizar a avaliação e a distribuição destes entre as diversas ações a serem realizadas. Portanto, mais especificamente, o plano engloba os métodos de avaliação e o cronograma das ações do avaliador. Da mesma forma que para as etapas anteriores, as atividades do plano de avaliação basear-se-ão no propósito da avaliação, além de apresentar diferenças significativas se o avaliador for alguém do desenvolvimento, um provável adquirente do produto ou alguém da equipe de certificação da qualidade. Para cada um desses, a norma especifica as atividades nas Partes 3, 4 e 5, respectivamente (ver em 3.3.2).

#### 3.3.2.4. Executar a avaliação

Após todos os requisitos pré-estabelecidos, as métricas selecionadas, os níveis de pontuação e critérios de julgamento especificados e o plano de ação projetado, pode-se de fato executar a avaliação. Para tanto são feitas as medições aplicando-se as métricas selecionadas ao produto de software, e tendo como resultado os valores nas escalas das métricas. Os valores medidos são então comparados com os critérios pré-determinados na especificação da avaliação. Por fim, é feito o julgamento do produto do software através de uma declaração de quanto este atende os requisitos de qualidade. A decisão da aceitação ou liberação deste produto dar-se-á baseada nesta averiguação de qualidade em conjunto com as características de eficiência em tempo e custo.

No Anexo 3 encontra-se um modelo de relatório de avaliação de produto de software que um avaliador pode apresentar como resultado documentado de seu trabalho.

Como referenciado na própria norma, o processo de avaliação descrito para produtos de software pode perfeitamente ser utilizado para verificar a conformidade de pacotes de software. A norma que especifica a qualidade para este tipo de produtos é apresentada, com menor ênfase, a seguir.

### **3.3.3. A Norma NBR ISO/IEC 12119**

Válida desde trinta de novembro de 1998, a norma NBR ISO/IEC 12119 – Tecnologia de Informação – Pacote de software – Teste e requisitos de qualidade, como o próprio título diz, trata da avaliação de pacotes de software, também conhecidos como “software de prateleira”. Como exemplo destes podem ser citados os processadores de texto, as planilhas eletrônicas, os softwares de gráficos, os bancos de dados etc, assim como eles estão disponíveis no mercado (ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS, 1998).

Essa norma estabelece detalhadamente os requisitos de qualidade que os pacotes devem obrigatoriamente respeitar, sugerindo ainda alguns requisitos opcionais (indicados no texto pela expressão “convém que”).

De acordo com a norma, o pacote de software a ser testado deve dispor de três requisitos de qualidade: uma descrição detalhada do produto, documentações dirigidas ao usuário e, claro, programas e dados fornecidos para a aplicação do produto de software. Cada um desses requisitos é apresentado sucintamente a seguir.

#### 3.3.3.1. Descrição de produto

Uma descrição do produto deve estar disponível a quem se interessar pelo produto; por exemplo, pode esta aparecer resumida na caixa do produto. Essa é extremamente necessária, pois ajuda o usuário ou o comprador na avaliação da adequação do produto às suas necessidades. Além disso, essa descrição pode ser usada como base para testes do pacote de software.

É essencial que cada declaração sobre o produto seja: inteligível, completa, bem organizada e apresentada, correta, testável e que seja livre de inconsistências internas.

Nessa descrição é importante constar:

- A identificação do produto (nome, versão e data);
- A identificação do fornecedor (nome e endereço);
- As tarefas que podem ser executadas com o pacote;
- Os requisitos de hardware e software;
- A possibilidade de interface com outros produtos, caso isso seja relevante;
- A listagem dos itens a serem entregues;
- O modo de instalação;
- A existência ou não de suporte técnico; e
- A oferta ou não de serviço de manutenção.

Na descrição do produto devem constar também declarações sucintas sobre as seis características de qualidade como definidas na NBR 13596: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

Quanto à funcionalidade é preciso estabelecer:

- Uma visão geral das funções disponíveis, os dados que serão necessários e as facilidades oferecidas;
- Os valores-limite, caso o uso do produto tenha, por exemplo, número de registros ou tamanho de amostra limitados; e
- Uma estratégia para segurança de acesso a programas e dados.

Quanto à confiabilidade é necessária a inclusão de informações sobre os procedimentos para preservação de dados.

Já sobre a usabilidade é importante ter especificado:

- O tipo de interface com o usuário;
- O conhecimento, técnico, operacional ou em idiomas, requerido para a aplicação do produto;
- A possibilidade de o usuário adaptar o produto a suas necessidades, bem como as informações associadas a essa adaptação;
- Quando da existência de proteção contra infrações e direitos autorais, se estas influenciam a usabilidade; e

- Dados sobre a eficiência de uso e satisfação do usuário, assim como apresentado na Parte 11 da norma ISO 9241, que será mencionada mais adiante nesse capítulo.

Quanto à eficiência, é interessante ter relatado na descrição do produto como este se comporta em relação ao tempo, principalmente no que diz respeito aos tempos de resposta do programa às ações do usuário.

Sobre as características de manutenibilidade e portabilidade, algumas informações podem ser importantes, principalmente aquelas que dizem respeito a possibilidade de alterações no produto e de transferência deste para outro ambiente.

#### 3.3.3.2. Documentação de usuário

Voltada totalmente para o usuário, esta deve proporcionar ao mesmo uma visão geral do produto, contendo todas as informações necessárias para o bom uso do mesmo. De acordo com a norma, todas as funções estabelecidas anteriormente na descrição de produto e todas as funções do programa que o usuário tiver acesso, devem estar aqui descritas corretamente e de forma detalhada.

Este documento deve ser completo, consistente e fácil de entender. Além disso, quando a instalação e manutenção puderem ser executadas pelo usuário, um manual contendo as informações necessárias para tal deve estar incluído neste.

Caso haja alguma limitação, assim como especificado na descrição de produto, esta também deve estar aqui estabelecida.

#### 3.3.3.3. Programas e dados

Na descrição de produto é dada uma visão geral das características do software de acordo com as características de qualidade definidas na NBR 13596. Nesta parte a descrição de tais características deve ser feita em detalhes, focando principalmente questões como:

- Cada uma das funções do software, descritas exatamente na forma em que são executadas;
- A uniformidade do controle das operações do programa pelo usuário;
- A inteligibilidade das perguntas, mensagens e respostas do programa; e

- A reversibilidade de funções que têm conseqüências graves.

Barreto (2000) resume em algumas recomendações os requisitos que o pacote de software deve contemplar segundo a norma. Nestas o autor identifica os seguintes elementos que devem estar presente no produto:

- Documentação do usuário de fácil compreensão;
- Um sumário e um índice remissivo da documentação do usuário;
- Possibilidade de verificar se uma instalação foi bem sucedida;
- Especificação de valores-limite para todos os dados de entrada, que deverão ser testados;
- Operação normal mesmo quando os dados informados estão fora dos limites especificados;
- Consistência de vocabulário entre as mensagens e a documentação;
- Função de auxílio (*help*) com recursos de hipertexto;
- Mensagens de erro com informações necessárias para a solução da situação de erro;
- Diferenciação dos tipos de mensagem: confirmação, consulta, advertência e erro;
- Clareza no formato das telas de entrada e relatórios;
- Capacidade de reverter funções de efeito drástico;
- Alertas claros para as conseqüências de uma determinada confirmação;
- Identificação dos arquivos utilizados pelo programa;
- Identificação da função do programa que está sendo executada no momento; e
- Capacidade de interromper um processamento demorado.

Por fim, a norma apresenta as instruções para serem feitos testes no pacote de software. Porém, como visto no item 3.3.2.4, este tipo de produto também pode ser avaliado seguindo os critérios estabelecidos na norma 14598.

Para encerrar o estudo da qualidade de software, segundo a série de normas ISO, faz-se necessário ainda tecer alguns comentários sobre a norma ISO 9241, que trata especificamente das exigências ergonômicas em trabalhos com terminais de vídeo.

### 3.3.4. A Norma ISO 9241

Apesar de não tratar especificamente de softwares, a norma internacional ISO 9241-*Ergonomic requirements for office work with visual display terminals* (Exigências ergonômicas para o trabalho de escritório com terminais de vídeo) pode contribuir significativamente para a análise da interface de um produto de software. A norma tem como principal objetivo “promover um bom design ergonômico para trabalho com terminais de vídeo e garantir que os usuários operem os monitores com segurança, eficiência, efetividade e conforto” (ISO 9241, 1992, p. iv – tradução livre). A norma, dividida em dezessete partes, apresenta uma introdução (Parte 1) e recomendações baseadas em estudos empíricos e revisão bibliográfica sobre: os requisitos da tarefa (Parte 2), o layout da estação de trabalho (Parte 5) e as características do ambiente de trabalho (Parte 6). Além disso, faz também recomendações específicas sobre a ergonomia do hardware (Partes 3, 4, 7, 8 e 9) e da interface do software (Partes de 10 a 17). Essas últimas, mais relevantes para esse estudo por tratarem especificamente do software, são assim intituladas:

- Parte 10: Princípios de diálogos
- Parte 11: Condução da usabilidade
- Parte 12: Apresentação da informação
- Parte 13: Condução do usuário
- Parte 14: Diálogos de menu
- Parte 15: Diálogos de comando
- Parte 16: Manipulação direta
- Parte 17: Diálogos de preenchimento de formulário

Especificamente na Parte 10 dessa norma encontram-se sugestões de como deve ocorrer a interação entre usuário e sistema (diálogo), para que seja possível alcançar um objetivo. A análise desta mostra que a mesma vem ao encontro das idéias de Scapin e Bastien (1997) descritas no capítulo anterior desse trabalho, evidenciando “a recorrência de definições e a estreita correlação entre os trabalhos de levantamentos de qualidades ergonômicas mínimas desejáveis para os produtos de software” (MEDEIROS, 1999, p. 77). As recomendações feitas nessa parte da norma são quanto: à adequação da tarefa, a auto-descrição, a controlabilidade, a conformidade com as expectativas do usuário, a tolerância a

erros, a adequação à individualização e a adequação ao aprendizado. Estas são apresentadas a seguir.

#### 3.3.4.1. Recomendações quanto à adequação da tarefa

O diálogo entre usuário e sistema pode ser considerado adequado à tarefa se o mesmo possibilita a execução das atividades pertinentes de maneira eficaz (alcance dos objetivos) e eficiente (ótima relação entre recursos empregados para o alcance dos objetivos). As seguintes recomendações são feitas:

- Ao usuário devem ser somente apresentadas informações pertinentes à execução da tarefa;
- As informações de ajuda (*help*) devem estar relacionadas à tarefa;
- Qualquer ação que possa ser realizada sem o envolvimento do usuário deve ser automatizada;
- O diálogo deve ser projetado de forma a considerar as habilidades e experiências do usuário;
- Os formatos de entrada e saída devem ser apropriados à tarefa em questão e aos requisitos do usuário;
- O diálogo deve dar suporte ao usuário na execução de tarefas recorrentes;
- Se há possibilidade de entradas *default* para uma tarefa, estas devem estar automaticamente indicadas;
- Deve ser possível substituir os valores *default* por outros valores apropriados;
- Durante a execução de uma tarefa onde os dados foram alterados, os dados originais devem permanecer acessíveis caso sejam necessários; e
- O diálogo deve evitar que o usuário seja obrigado a executar passos da tarefa desnecessários.

#### 3.3.4.2. Recomendações quanto à auto-descrição

O diálogo pode ser considerado auto-descritivo se as explicações e *feedbacks* são fácil e rapidamente interpretadas pelo usuário. Para tanto são feitas as seguintes recomendações:

- O diálogo deve proporcionar *feedback* apropriado ao final de qualquer ação, fornecendo explicações e pedindo confirmação principalmente se a ação tiver conseqüências significativas;

- A terminologia utilizada deve ser apropriada e relativa ao ambiente da tarefa e das características do usuário;
- *Feedbacks* e explicações devem ser disponibilizados ao usuário no decorrer da tarefa, para que o mesmo possa ter um entendimento geral do sistema de diálogo;
- *Feedbacks* e explicações devem considerar o nível de conhecimento do usuário típico, e podem ser apresentados em diferentes tipos e tamanhos, de acordo com as necessidades do usuário;
- A qualidade dos *feedbacks* e das explicações deve minimizar a necessidade de consulta a manuais e outras informações externas;
- Se existirem *defaults* para uma tarefas, estes devem estar indicados ao usuário;
- O usuário deve ser informado sempre que houver mudança no estado do sistema de diálogo (por exemplo, alteração da mensagem “Procurando” para “Encontrado”);
- O sistema de diálogo deve dar informações quanto ao que ele considera um valor possível de uma entrada (por exemplo, informar como é o campo “Data”); e
- As mensagens devem ser objetivas, consistentes e construtivas, evitando qualquer juízo de valor.

#### 3.3.4.3. Recomendações quanto à controlabilidade

O diálogo pode ser considerado controlável se parte do usuário a inicialização da interação, a determinação da direção e o ritmo dos trabalhos. As seguintes características são sugeridas:

- O usuário deve ter o controle da velocidade de interação;
- O usuário deve ter o controle da continuidade do diálogo;
- Caso a tarefa permita, o usuário deve poder interromper o diálogo e, quando resolver retomá-lo, poder iniciar do ponto onde havia parado;
- Caso a tarefa permita, o usuário deve poder desfazer, pelo menos, um passo do diálogo;
- O diálogo deve considerar, para seu controle, as diferentes necessidades e características dos usuários em termos de interação;
- O usuário deve ser capaz de controlar a quantidade de dados apresentados, caso isso seja relevante para a execução da tarefa; e
- O usuário deve ter o controle sobre como representar as entradas e saídas do sistema e, se possível, deve poder decidir qual o dispositivo mais adequado para tal.



#### 3.3.4.4. Recomendações quanto à conformidade com as expectativas do usuário

O diálogo pode ser considerado em conformidade com as expectativas do usuário se o mesmo respeitar as características deste quanto a sua experiência, seu grau de instrução e conhecimento. As recomendações da norma para este princípio são:

- O diálogo deve ser consistente, em relação a seu comportamento e aparência, em todo o sistema;
- Ações de mudança de estado devem ser implementadas consistentemente;
- O vocabulário usado no decorrer da execução da tarefa deve ser familiar ao usuário;
- Em tarefas similares, o diálogo usado deve ser similar;
- O *feedback* imediato às entradas do usuário deve atender às expectativas do usuário e estar de acordo com o nível de conhecimento do mesmo;
- O cursor deve aparecer no local onde a entrada é necessária; e
- O usuário deve ser informado se o tempo de resposta de uma atividade específica for consideravelmente diferente do esperado.

#### 3.3.4.5. Recomendações quanto à tolerância a erros

O diálogo pode ser considerado tolerante a erros se possibilita que o usuário obtenha os resultados desejados ao mesmo tempo que minimiza a necessidade de correções de erros. Para tanto a norma recomenda o seguinte:

- Ao usuário deve ser dado apoio para a detecção e prevenção de erros de entrada, principalmente aqueles que causarem falhas do sistema;
- Os erros devem ser explicados de forma que instrua o usuário para sua correção;
- Dependendo da tarefa, pode ser desejável aplicar esforço especial nas técnicas de apresentação para melhorar o reconhecimento de situações de erro e respectivas recuperações;
- Caso o sistema de diálogo possa corrigir erros automaticamente, isto deve ser enfatizado ao usuário, mostrando sempre o que foi alterado;
- Caso seja importante, situações de erro devem ser adiadas, deixando ao usuário a decisão de quando e como tratá-las;
- Os dados devem ser verificados e validados antes de serem processados e, para comandos com sérias consequências, devem ser proporcionados controles adicionais como confirmação;

- Caso o usuário solicite, explicações adicionais devem ser dadas durante a correção de erros; e
- Quando a tarefa permitir, a correção de erros deve ser possível, sem que, para isso, seja necessária a troca de estados do sistema de diálogo.

#### 3.3.4.6. Recomendações quanto à adequação à individualização

O diálogo pode ser considerado adequado à individualização se for possível fazer mudanças na interface de forma a atender as necessidades da tarefa, as preferências individuais e experiências do usuário. As sugestões da norma neste caso são:

- O diálogo deve poder ser adaptado à cultura e linguagem do usuário, seu conhecimento e experiências na realização da tarefa, suas habilidades cognitivas, senso-motoras e perceptivas;
- O sistema de diálogo deve permitir que o usuário escolha formas de representação alternativas de acordo com suas preferências individuais e com a complexidade da informação a ser processada;
- A quantidade de explicações deve ser diferenciada de acordo com o nível individual de conhecimento do usuário;
- Na medida do possível o usuário deve poder incorporar seu vocabulário na determinação de nomes de objetos e ações, podendo também incluir comandos personalizados;
- O usuário deve ser capaz de configurar parâmetros de tempo operacional de acordo com suas necessidades individuais; e
- Os usuários devem ser capazes de escolher diferentes técnicas de diálogo para diferentes tarefas.

#### 3.3.4.7. Recomendações quanto à adequação ao aprendizado

O diálogo pode ser considerado adequado ao aprendizado se o mesmo suporta e guia o usuário na aprendizagem do uso do sistema. As seguintes recomendações são feitas para esse princípio:

- As regras e conceitos que sejam úteis para o aprendizado devem estar disponíveis ao usuário;
- Estratégias de aprendizado relevantes devem estar disponíveis ao usuário (por exemplo, tutoriais interativos);

- Facilidades de reaprendizado, como atalhos, devem ser oferecidas; e
- Ao usuário devem ser proporcionados diferentes meios para torná-lo familiarizado com os elementos de diálogo.

Apesar da existência de diversas metodologias de avaliação ergonômica e normas de qualidade, a escolha de um software não é uma tarefa fácil. As pessoas em geral não conhecem os termos técnicos usados na descrição do produto como também pouco pensam em questões quanto à funcionalidade e adequação a suas preferências e à tarefa a ser executada. A forma mais adequada para decidir quanto a aquisição de um software seria iniciar com uma experimentação do mesmo. Durante a navegação o usuário pode verificar se os atributos oferecidos pelo software são adequados e suficientes para atingir seus objetivos.

Conhecer as preferências dos usuários pode ser um caminho interessante a seguir quando se deseja desenvolver ou avaliar um software. Técnicas que têm se mostrado bastante eficazes para a obtenção de informações relevantes sobre o comportamento do usuário são as Técnicas de Preferência Declarada, as quais serão apresentadas no próximo capítulo.

## **CAPÍTULO 4 – TÉCNICAS DE PREFERÊNCIA DECLARADA**

### **4.1. Introdução**

Todo indivíduo, quando solicitado a escolher um produto ou serviço entre vários outros, acaba por, de alguma forma, revelar suas preferências. Em alguns casos, é possível identificar claramente os atributos que o levaram a selecionar tal produto. Além disso, também na maioria dos casos, é possível relacionar as características do produto escolhido com o perfil do indivíduo. Desse modo, quando o cliente está para adquirir um produto ou optar por um determinado serviço, é possível identificar sua faixa etária, sua renda aproximada e seu grau de instrução; da mesma forma é possível obter as características do bem ou serviço que o levaram a ser preferido pelo cliente. Uma vez efetuada a compra, os dados refletem o comportamento real do usuário perante as diversas opções disponíveis.

Porém, avaliar o comportamento real do usuário pode não ser suficiente para analisar os fatores que determinaram suas escolhas. Isso porque, em muitos casos, a pessoa adota um certo comportamento simplesmente por falta de opção. Além disso, devido às muitas variáveis relevantes, fica difícil entender e prever o comportamento dos consumidores e os *trade-offs* realizados em suas tomadas de decisão. Essa noção de valor que as pessoas dão a questões como preço, tempo e qualidade, podem ser obtidas através de técnicas de Preferência Declarada.

Estas técnicas têm como objetivo principal obter informações sobre as preferências das pessoas e suas possíveis ações, a partir da arguição sobre o que elas preferem ou o que fariam em uma dada situação. A aplicação dessas técnicas consiste em oferecer aos entrevistados um conjunto de opções de escolha pré-estabelecido de forma a refletir os *trade-offs* realizados pelos mesmos. Questões como, por exemplo “Quanto as pessoas estão dispostas a pagar por um certo ganho de tempo num serviço de transporte?” e “O que é mais relevante em certo sistema computacional, velocidade ou precisão?”, podem ser respondidas mesmo sem que as pessoas de fato experimentem tais realidades.

Técnicas de preferência declarada “tradicionalis” proporcionam aos entrevistados um conjunto de alternativas que eles devem ordenar ou avaliar cada uma individualmente com

uma nota, de acordo com suas preferências. Já em experiências de escolha declarada, é oferecido aos entrevistados um conjunto de alternativas, e estes devem expressar sua escolha indicando uma única alternativa ou nomeando probabilidades a cada uma delas. Em qualquer um desses casos a técnica faz uso de “qualquer pesquisa que trata do comportamento esperado ao invés do atual” (JONES, 1991, p. 1).

As técnicas de preferência declarada surgiram a partir da teoria da demanda do consumidor e foram amplamente aplicadas em marketing na década de setenta. Até hoje essas técnicas vêm sendo aplicadas em áreas como psicologia, sociologia e planejamento urbano. Mais especificamente no planejamento e análise de serviços de transporte, vários trabalhos tem sido feitos utilizando as técnicas de preferência declarada.

Senna (1992) usou a preferência declarada para estimar o valor de reduções na variabilidade do tempo de viagem para diferentes destinos e diferentes restrições de tempo de chegada. Os resultados encontrados com essa abordagem sugerem que o benefício de mais viagens regulares é superior aos benefícios obtidos com a redução da variabilidade em tempo de viagens definidos em estudos anteriores.

Novaes e Carvalho (1994) desenvolveram um modelo de divisão de mercado do sistema de transporte rodoviário de passageiros na linha Florianópolis – Blumenau onde na função utilidade eram considerados fatores como tarifa, tempo de viagem, nível de conforto e renda do passageiro. Elasticidade e taxas marginais de substituição foram calculadas e analisadas.

Bastos (1994) utilizou as técnicas de preferência declarada em sua tese de doutorado para o planejamento da rede escolar, visando identificar os melhores locais para o estabelecimento de unidades escolares. Numa linha similar, Almeida (1999) desenvolveu uma metodologia que pode ser utilizada como instrumento auxiliar no planejamento de sistemas educacionais a partir da incorporação de aspectos comportamentais dos indivíduos através da utilização de técnicas de preferência declarada. Resultados quanto a aplicação dessa metodologia foram relatados em Almeida e Gonçalves (1999, 2000 e 2001).

Souza (1999) usou as técnicas de preferência declarada para modelar o comportamento dos consumidores da cidade de Guarapuava, localizada na região centro sul do estado do

Paraná, quanto a escolha do local que efetuarão suas compras mensais. Uma contribuição adicional de seu trabalho de doutorado foi o desenvolvimento de um software que, a partir do método da máxima verossimilhança, obtém boas estimativas para os parâmetros da função matemática que representa a utilidade.

Schmitz (2001) fez uso da preferência declarada no desenvolvimento de uma metodologia que apóia as discussões quanto à avaliação do potencial econômico de uma rodovia. Com as pesquisas de preferência declarada o autor consegue verificar a disposição dos usuários em pagar uma tarifa de pedágio, sendo que essa conclusão pode sinalizar quanto à viabilidade econômica do empreendimento. A metodologia proposta foi aplicada na avaliação da disponibilidade dos usuários da rodovia SC 401 em pagar pedágio. Os resultados obtidos com a pesquisa mostraram que a metodologia desenvolvida pelo autor foi bastante prática e consistente.

Vários outros trabalhos (LOUVIERE, 1988; ORTUZAR e WILLUMSEN, 1994; NOVAES *et al*, 1996; CONSTANTINO, 1997; CAMARGO, 2000; BHAT e CASTELAR, 2002; MARQUES, 2003) fazem uso das técnicas de preferência declarada para estimar as preferências dos indivíduos. No presente capítulo será relatado como proceder para a aplicação de um experimento utilizando estas técnicas.

#### **4.2. Conceituação da Preferência Declarada**

Para uma análise comportamental das decisões dos indivíduos, duas abordagens distintas podem ser realizadas: uma é a observação direta das ações desses indivíduos, a outra é a arguição dos mesmos quanto a suas preferências. À primeira técnica dá-se o nome de Preferência Revelada (*revealed preference*) e à segunda Preferência Declarada (*stated choice* ou *stated preference*).

A preferência revelada (PR) identifica as decisões tomadas pelos indivíduos a partir de questionamentos ou observações diretas que levantam as ações realmente efetivadas. A principal limitação desta técnica comportamental é que só é possível analisar cenários já existentes, tornando impossível a identificação do comportamento dos indivíduos em situações inéditas. Desse modo, essa técnica nem sempre é adequada para o planejamento ou a implantação de novos serviços. Além disso, Kroes e Sheldon (1988, p. 12) argumentam que:

- Pode ser difícil obter variações suficientes nos dados da preferência revelada para estimar todas as variáveis de interesse;
- Há geralmente fortes correlações entre as variáveis de interesse, o que pode dificultar a análise individual de cada atributo; e
- A técnica requer que as variáveis sejam expressas em unidades objetivas, sendo difícil analisar variáveis subjetivas.

As técnicas de preferência declarada (PD), também designadas de análise conjunta, surgiram com o intuito de suprir tais deficiências. Uma definição mais formal destas foi dada em 1978 por Green e Srinivasan (*apud* KROES e SHELDON, 1988, p. 12), como sendo “qualquer método decomposicional que estima a estrutura das preferências de consumidores dadas suas avaliações globais de um conjunto de alternativas que são pré-especificadas em termos dos níveis dos diferentes atributos.” É considerado um método decomposicional aquele que, seguindo uma regra de composição, estima um conjunto de parcelas significativas para os atributos a partir de uma avaliação subjetiva das alternativas.

Mais especificamente na área de pesquisa em transportes, as técnicas de PD são definidas como “uma família de técnicas que utilizam declarações dos entrevistados sobre suas preferências em um conjunto de opções para estimar as funções utilidade.” (KROES e SHELDON, 1988, p.11). Almeida (1999, p. 61) afirma que

essas técnicas permitem aos pesquisadores que estudem e modelem aspectos do comportamento dos usuários relativos à avaliação e escolha dentre os produtos e serviços disponíveis para os mesmos.

Assim sendo, as técnicas de preferência declarada são na realidade ferramentas que apresentam aos entrevistados um conjunto de opções, que são cenários reais ou hipotéticos, construídos pelo pesquisador, que consideram os principais fatores que influenciam a escolha. O entrevistador apresenta as alternativas viáveis, cada qual representada por um conjunto de atributos que identificam o bem ou serviço. Ao entrevistado é feita uma pergunta do tipo “Se você tiver essas alternativas disponíveis, qual delas você escolhe?”. Souza (1999, p. 17) argumenta que

um consumidor, em princípio, escolhe um produto ou serviço pelo conjunto de atributos que o produto possui, obedecendo a uma escala subjetiva de valor para cada atributo, em relação ao seu custo. Assim, o consumidor

busca um produto ou serviço cujo somatório dos atributos seja igual ou superior ao valor disposto a pagar.

Isso significa que os indivíduos sempre selecionam a opção que maximiza a sua utilidade pessoal.

As respostas dos entrevistados quanto às suas preferências podem ser informadas de três modos: através de uma única opção, considerada pelo indivíduo como a melhor de todas (*choice*); através da ordenação decrescente das alternativas apresentadas (*ranking*) ou pela classificação, através de uma escala de avaliação (*rating*).

Em geral as técnicas de preferência declarada têm as seguintes características:

- são fáceis de controlar, pois é o pesquisador que define as condições que estão sendo avaliadas pelo entrevistado;
- são flexíveis, sendo capazes de lidar com uma ampla gama de variáveis, inclusive com atributos que não podem ser facilmente quantificados e
- são mais baratas de aplicar que as de preferência revelada, já que cada indivíduo entrevistado propicia múltiplas observações sobre variações das variáveis de interesse para a análise. (KROES e SHELDON, 1988, p.11).

Há uma tendência na adoção de métodos híbridos de pesquisa, utilizando tanto as técnicas de preferência declarada quanto as de preferência revelada (BEN-AKIVA e MORIKAWA, 1990; BHAT e CASTELAR, 2002). Jones (1991, p. 2) comenta que resultados obtidos com a preferência declarada têm se mostrado satisfatórios para prever o comportamento escolhido pelas pessoas, especialmente quando são combinados com alguma informação de suas preferências reveladas.

#### **4.3. O Experimento de Preferência Declarada**

É praticamente unânime entre os pesquisadores da área que algumas questões devem ser analisadas no desenvolvimento de um experimento de preferência declarada. Estas incluem:

- A forma e complexidade do experimento;
- A medição da escolha;



- A escolha do método de entrevista;
- A seleção da amostra; e
- A análise dos dados.

As considerações sobre cada um desses aspectos são feitas a seguir.

#### **4.3.1. Forma e Complexidade do Experimento**

Com relação à forma e complexidade do experimento, primeiramente, deve-se decidir quais serão os atributos e em quantos níveis cada um deve se dividir. Os atributos podem ser racionais, que são fatores que representam as características sócio-econômicas dos indivíduos e subjetivos, que são aqueles fatores não expressos diretamente a partir de conceitos econômicos, como, por exemplo, conforto e segurança. Os níveis nada mais são do que graduações que esses atributos podem contemplar.

Quanto maior o número de atributos, maior o número de fatores presentes num experimento e quanto mais níveis são considerados para cada atributo, mais detalhada fica a avaliação. Entretanto, quanto maior o número de atributos e níveis, maior será a complexidade da tarefa dos entrevistados, podendo isso inclusive inviabilizar o experimento.

O número total de alternativas e conseqüentemente de diferentes cenários vai depender do número de atributos e seus respectivos níveis. Segundo Jones (1991, p. 8) na prática trabalha-se com um conjunto de sete a oito atributos com quatro a cinco níveis cada, resultando em no máximo quarenta estímulos a serem representados. A consideração de todos os atributos e níveis é conhecida como um experimento fatorial completo.

De acordo com Schmitz (2001, p. 86) a combinação de todos os níveis e atributos num experimento estatístico segundo um fatorial completo tem demonstrado na prática, que na maioria das vezes o número total de combinações torna-se impraticável e de difícil avaliação. A experiência tem mostrado que existe uma quantidade limite de opções de escolhas a serem oferecidas ao entrevistado. Uma das alternativas é o desenvolvimento de projetos fracionados. Neste caso somente uma seleção de todas as combinações possíveis é apresentada aos entrevistados. Se mesmo assim o número de alternativas for grande, pode-se dividir este em conjuntos menores, sendo que faz-se necessário que em cada um desses subconjuntos esteja

incluída uma alternativa comum. Um experimento que faz uso dessa metodologia é conhecido como fatorial fracionado em blocos. Jones (1991, p. 8) alerta para que no design desse tipo de experimento fatorial fracionado seja preservado o grau de independência dos atributos, reduzindo assim problemas de colinearidade.

A maneira de apresentar ao entrevistado os atributos e seus respectivos níveis também é de suma importância. Um método considerado bastante eficaz e que é muito utilizado é a representação das alternativas através de cartões. Neste caso deve-se dar atenção à qualidade gráfica destes cartões, incluindo, sempre que possível, figuras representativas que auxiliem a compreensão do usuário.

#### **4.3.2. Medição da Escolha**

Nessa etapa define-se como os entrevistados podem registrar suas preferências, sendo as formas mais usadas as seguintes:

- Ordenação (*ranking*): neste caso é apresentado ao entrevistado um grupo de alternativas que este deve ordenar segundo suas preferências, estando implícita uma hierarquia de valores de utilidade.
- Avaliação (*rating*): neste método os entrevistados são solicitados a expressar seus graus de preferências para uma determinada opção utilizando uma escala numérica, normalmente entre 1(um) e 10 (dez).
- Escolha Declarada (*stated choice*): consiste basicamente em solicitar ao entrevistado que escolha apenas uma dentre as várias opções que lhe são oferecidas dentro de cada bloco.

A escolha do método de medição a ser adotado está diretamente relacionada ao método de análise de dados que será utilizado.

#### **4.3.3. Escolha do Método de Entrevistas**

Pode-se obter as preferências dos entrevistados de diferentes maneiras. As mais usuais são entrevista face a face e questionários auto-explicativos. É possível também a adoção de um método híbrido que em geral consiste no envio de material explicativo por correio ou meio eletrônico e em uma entrevista conduzida por telefone.

As entrevistas face a face, apesar de serem mais caras, são as que geram melhores resultados. Isso porque o entrevistador pode explicar diretamente os objetivos requeridos com a entrevista, além de poder gerenciar os *trade-offs*, estes moldados de acordo com as circunstâncias. Kroes e Sheldon (1988, p. 16) afirmam que quando as entrevistas são bem estruturadas e conduzidas por pessoal experiente garantem que:

- O conhecimento do processo de avaliação do entrevistado seja completamente entendido pelo pesquisador;
- O entrevistado não seja estimulado por fatores externos;
- Sejam proporcionadas as especificações completas das alternativas, de modo que os indivíduos as percebam como um todo e não como uma soma de um número de fatores separados.

Os questionários auto explicativos são usados mais freqüentemente no caso de ser solicitada a escolha de uma única alternativa. Apesar de mais baratos e de maior cobertura espacial, não há com a adoção desses uma garantia de retorno das respostas já que é o entrevistado que envia de volta, por correio ou meio eletrônico, o questionário preenchido. Ainda, nestes casos, não há como ter garantias quanto à veracidade e qualidade das respostas recebidas.

A adoção de métodos híbridos é bastante interessante, haja visto que, com o recebimento antecipado do material explicativo, o entrevistado pode refletir sobre suas escolhas e fazer melhores julgamentos antes de efetivar suas respostas. Caso haja alguma dúvida ou incerteza, essa pode ser esclarecida no momento da entrevista por telefone.

#### **4.3.4. Seleção da Amostra**

Segundo Ben-Akiva e Lerman (1985) o primeiro passo está em definir a população agregada considerada relevante, pois, para que o experimento tenha validade é necessário que esta seja representativa.

É importante salientar que a amostra deve ser escolhida aleatoriamente ou em um grupo que esteja diretamente envolvido no processo. A segunda forma é mais eficiente e desse modo os integrantes da amostra, no caso de uma eventual necessidade, são mais fáceis de

contatar; porém, há de se tomar cuidado para que uma amostra tomada dessa maneira não seja tendenciosa.

De qualquer forma os integrantes da amostra não precisam necessariamente ter vivenciado as opções apresentadas. No entanto, para maximizar o realismo do experimento, os entrevistados devem conseguir vislumbrar a situação apresentada o máximo possível.

#### 4.3.5. Análise dos Dados

A análise dos dados é uma etapa muito importante dentro da técnica de preferência declarada. As técnicas de estimação de parâmetros mais utilizadas são:

- Análise de regressão múltipla, que requer dados de avaliação, os quais são usados como variáveis dependentes sendo os atributos e níveis tratados como variáveis independentes. O modelo estima uma função utilidade que melhor explica o conjunto de escolhas feitas. Como resultado final, obtém-se a importância relativa de cada atributo. Bastos (1994, p. 25) cita os métodos de mínimos quadrados ordinários, ponderados ou generalizados como os mais utilizados para estimação neste caso.
- Modelo logit multinomial, que é o método mais utilizado para estimar os modelos de escolha a partir de dados de preferência declarada e que pode ser aplicado a qualquer tipo de medida de preferência, seja avaliação, ordenação ou escolha declarada. Segundo Jones (1991, p. 10) duas abordagens são possíveis neste caso: a comparação par a par e o modelo logit explodido que adota a ordenação de dados.

Também faz-se uso de programas computacionais com a finalidade de ajustar a função utilidade, como por exemplo, o *software* LMPC que utiliza o método de máxima-verossimilhança para obter as estimativas dos parâmetros (SOUZA, 1999).

Como os experimentos baseados em PD tratam geralmente de situações hipotéticas, é importante desenhá-los da maneira mais realística possível, cuidando principalmente com o contexto em que se insere a situação, as opções apresentadas para análise e as respostas possíveis. Fowkes e Wardman (1988) sugerem que os experimentos sejam baseados em atributos com apenas poucos níveis e que os valores possíveis para estes sejam razoavelmente realísticos, tentando dessa forma customizar o design. Eles alertam que

se o design dos experimentos com PD apresentar pequenas diferenças nos atributos, elas podem ser ignoradas por alguns respondentes se outros atributos variarem entre as opções a serem comparadas (FOWKES e WARDMAN, 1988, p. 34).

#### **4.4. A Função Utilidade**

Como comentado anteriormente, um consumidor escolhe um bem ou serviço pelo conjunto dos atributos que este possui, buscando aquele com melhores características de forma a mais eficientemente satisfazer seus objetivos. A esta atratividade do bem ou serviço em termos de seus atributos dá-se o nome de utilidade.

Segundo Schmitz (2001, p. 81) a utilidade “é um valor que um determinado indivíduo pode atribuir a um produto ou serviço através de uma combinação de fatores tal que esse valor seja máximo para a escolha realizada dentro de um conjunto de opções”. Matematicamente a função utilidade pode ser expressa por uma única função objetivo que expressa a atração em termos de seus atributos, função essa que o indivíduo busca maximizar em seu processo decisório (BEN-AKIVA e LERMAN, 1985, p. 37). Os valores desta função utilidade permitem que se estabeleça uma comparação entre a utilidade dos diferentes atributos considerados relevantes.

Em preferência declarada a função utilidade de uma alternativa tem como objetivo decompor as preferências admitidas dos entrevistados em partes de utilidades referentes a cada atributo selecionado do sistema em estudo. Assim, pode-se estabelecer o efeito relativo de cada atributo na utilidade total. Esta função é normalmente representada por um modelo linear aditivo cuja forma matemática é dada por:

$$U_{in} = \beta_1 x_{in1} + \beta_2 x_{in2} + \dots + \beta_k x_{ink} = \sum_{k=1}^K \beta_k X_{ink} \quad (4.1)$$

onde:

$U_{in}$  é a utilidade da alternativa  $i$  para o indivíduo  $n$ ;

$X = [X_{in1}, X_{in2}, \dots, X_{ink}]$  é o vetor dos valores dos  $k$  atributos da alternativa  $i$  para o indivíduo  $n$ ;

$\beta = [\beta_1, \beta_2, \dots, \beta_k]$  é o vetor dos  $k$  parâmetros desconhecidos que devem ser calibrados; e  $k$  é a quantidade de atributos das alternativas.

Em uma função utilidade devidamente ajustada, os coeficientes podem ser usados, por exemplo, para:

- Determinar a importância relativa dos atributos incluídos na experiência;
- Determinar valores de tempo ou monetários; e
- Especificar a função utilidade para modelos de previsão.

A função utilidade, seguindo o modelo matemático citado acima, deve respeitar as premissas de;

- Consistência, onde, sob condições idênticas, isto é, para o mesmo vetor  $X_{ink}$ , a probabilidade de escolha de qualquer alternativa não deve ser alterada; e
- Transitividade, onde comparando através da utilidade as alternativas  $i$  e  $j$  e posteriormente as alternativas  $j$  e  $k$ , se a alternativa  $i$  for melhor que a alternativa  $j$  e a alternativa  $j$  for melhor que a alternativa  $k$  então, necessariamente, a alternativa  $i$  deve ser melhor que a alternativa  $k$ .

A função utilidade definida pela equação (4.1) representa uma forma simplificada para modelagem da utilidade, pois existem influências na escolha do indivíduo que não são medidas ou reconhecidas pelos pesquisadores. Além disso, podem existir inconsistências nas respostas do indivíduo que levam inclusive a uma desconsideração da sua opinião para o processo de avaliação de alternativas. Desse modo, para superar essas limitações, deve ser adicionado à expressão determinística da equação (4.1) um elemento estocástico que representa o efeito dos fatores omissos. Assim, a função utilidade randômica pode ser expressa como a soma de uma componente determinística, relativa aos dados observados e, uma componente aleatória relativa aos elementos não observados, tendo então a expressão:

$$U_{in} = V_{in} + \varepsilon_{in} \quad (4.2)$$

onde:

$U_{in}$  é a utilidade da alternativa  $i$  para o indivíduo  $n$ ;

$V_{in}$  é a parcela determinística da alternativa  $i$  para o indivíduo  $n$ ; e  
 $\varepsilon_{in}$  é a parcela aleatória associada à alternativa  $i$ .

Segundo Ben-Akiva e Lerman (1985) as principais causas da aleatoriedade na função utilidade são relativas a atributos não observados, variações de comportamento não observadas e às informações imperfeitas.

Assumindo que o indivíduo seleciona a alternativa com maior utilidade em seu ponto de vista, na comparação de duas alternativas  $i$  e  $j$  através de suas funções utilidade  $U_i$  e  $U_j$  respectivamente, pode escolher aquela cuja utilidade seja maior, independentemente da diferença entre as duas utilidades. Contudo, devido à aleatoriedade da função utilidade, seria interessante o mesmo optar por uma escolha probabilística. Essa pode se dar da seguinte forma: define-se  $C_n$  o conjunto de alternativas apresentadas ao indivíduo  $n$ ; assim, a probabilidade de escolha da alternativa  $i$  é igual a probabilidade da utilidade da alternativa  $i$  ser maior ou igual a utilidade de todas as outras alternativas no conjunto de escolha  $C_n$ . Pode-se então escrever:

$$\begin{aligned} P(i / C_n) &= P[U_{in} \geq U_{jn}, \forall j \in C_n] \\ &= P[V_{in} + \varepsilon_{in} \geq V_{jn} + \varepsilon_{jn}, \forall j \in C_n] \\ &= P[V_{in} - V_{jn} \geq \varepsilon_{jn} - \varepsilon_{in}, \forall j \in C_n] \end{aligned} \quad (4.3)$$

A parcela determinística da utilidade é definida, como já mencionado, em função do conjunto de atributos da alternativa, tendo freqüentemente a forma matemática de uma função linear aditiva. Portanto, a parcela determinística da equação (4.2) satisfaz a seguinte expressão:

$$V_{in} = \sum_{k=1}^K \beta_k X_{ink} \quad (4.4)$$

Para modelar esta parcela determinística, é necessário que o pesquisador identifique todas as variáveis e determine como estas se combinam para influenciar na decisão. O problema na mensuração da função utilidade está então na avaliação da parcela probabilística  $\varepsilon_{jn} - \varepsilon_{in}$ , pois, de acordo com a expressão (4.3), o pesquisador há de comparar se esta é

maior, igual ou menor que a diferença determinística  $V_{in} - V_{jn}$ . A solução então está na identificação da distribuição de probabilidade dessa parcela aleatória. Uma variedade de modelos de probabilidade discreta podem ser formulados dependendo da distribuição de probabilidade assumida. Uma forma bastante adequada é assumir que os termos  $\varepsilon_{in}$  são independentes e identicamente distribuídos segundo uma função distribuição de probabilidade Gumbel (Apêndice A). Desse modo, as probabilidades podem ser determinadas por um modelo de escolha discreta conhecido como Modelo Logit Multinomial (BEN-AKIVA e LERMAN, 1985; SENNA, 1992; ORTÚZAR, 1999).

Assumindo então que os componentes estocásticos da função utilidade são idêntica e independentemente distribuídos segundo a distribuição de probabilidade Gumbel, tem-se que a probabilidade que o indivíduo  $n$  escolha a alternativa  $i$  em detrimento das demais é dada por:

$$P_n(i) = \frac{e^{\mu V_{in}}}{\sum_{j=1}^m e^{\mu V_{jn}}} \quad (4.5)$$

onde:

$P_n(i) = P(i / C_n)$  é a probabilidade da alternativa  $i$  ser escolhida pelo indivíduo  $n$  considerando um conjunto de opções  $C_n$ .

Na prática normalmente adota-se  $\mu = 1$ .

Quando o número de alternativas é igual a dois, temos o caso particular, denominado modelo logit binomial.

O método de calibração dos parâmetros do modelo mais freqüentemente utilizado baseia-se no procedimento estatístico da máxima verossimilhança (*maximum likelihood*). Esse ajuste objetiva, a partir de uma amostra, obter estimativas para os parâmetros de tal forma que a amostra observada tem máxima semelhança com o evento ocorrido. Em outras palavras, deseja-se determinar os componentes do vetor  $\beta$  que maximize a função verossimilhança que, para uma amostra de  $N$  observações, pode ser escrita conforme a expressão seguinte:



$$L(\beta) = \prod_{n=1}^N f(X_n, \beta) \quad (4.6)$$

onde:

$f(X_n, \beta)$  representa a função densidade de probabilidade;

$X_n = [X_1, X_2, \dots, X_n]$  é um conjunto de valores e

$\beta = [\beta_1, \beta_2, \dots, \beta_n]$  é o conjunto de parâmetros a estimar.

Quando o método de pesquisa de preferência declarada envolver o ordenamento de alternativas em um conjunto de opções, é comum adotar o modelo logit explodido (BEN-AKIVA e LERMAN, 1985). Neste caso há um desdobramento do processo de escolha em  $J-1$  decisões independentes e hierárquicas por parte de cada entrevistado, como se fossem escolhas feitas por diferentes entrevistados. Se as probabilidades de escolha seguem o mesmo modelo logit, a probabilidade de observar uma ordem de classificação para a alternativa 1 ser preferida a 2, a alternativa 2 ser preferida a 3, e assim por diante, é dada pelo produto de  $J-1$  funções do modelo logit multinomial, ficando a função verossimilhança como:

$$L^*(\beta) = \prod_{n=1}^N \prod_{i=1}^{J-1} \frac{e^{\beta X_{in}}}{\sum_{j=1}^J e^{\beta X_{jn}}} \quad (4.7)$$

Para determinação dos parâmetros utiliza-se a função log-verossimilhança:

$$L^*(\beta) = \sum_{i=1}^{J-1} \sum_{n=1}^N [\beta X_{in} - \ln \sum_{j=1}^J e^{\beta X_{jn}}] \quad (4.8)$$

No desenvolvimento do software LMPC - Logit Multinomial com Probabilidade Condicional para a maximização da função utilidade a partir da função log-verossimilhança, Souza (1999) utiliza-se do método de Newton-Raphson e da análise das derivadas primeira e segunda para obtenção dos parâmetros em questão. Além disso, testes estatísticos são

aplicados para verificar se os resultados obtidos são significativos e assintoticamente convergentes.

Conforme o que foi aqui apresentado, pode-se concluir que o principal benefício obtido com a utilização de técnicas de preferência declarada é a habilidade de captar resposta para as diversas combinações de atributos que em geral não são facilmente observados. Contudo, para a obtenção de bons resultados, deve-se ter certos cuidados em desenvolver experimentos os mais realistas possíveis. Jones (1991, p. 11) sugere algumas atitudes que podem ser tomadas quanto ao contexto dos exercícios, às opções que são apresentadas e às respostas permitidas:

- Focar o comportamento específico ao invés do geral;
- Usar um contexto de escolha real no qual o entrevistado seja experimentado ou se sinta inserido;
- Usar níveis de atributos existentes ou pelo menos possíveis de realização;
- Incorporar, quando possível, a verificação das respostas dadas pelos entrevistados;
- Garantir que todas as variáveis relevantes estejam incluídas na análise;
- Simplificar a apresentação da escolha das opções;
- Garantir que as restrições de escolha tenham sido consideradas; e
- Permitir que os entrevistados optem por uma resposta fora do conjunto de alternativas experimentais.

As técnicas de preferência declarada aqui apresentadas serão usadas, juntamente com as noções de qualidade e de ergonomia de software, para o desenvolvimento de uma metodologia de avaliação de software interativo, a qual será apresentada no capítulo seguinte.

## **CAPÍTULO 5 – METODOLOGIA DE AVALIAÇÃO PROPOSTA**

### **5.1. Introdução**

A metodologia de avaliação proposta foi elaborada a partir da revisão dos modelos de avaliação ergonômica e de qualidade de software apresentados nesse trabalho. Porém, esses modelos apresentaram-se consideravelmente rígidos, principalmente no que diz respeito à não consideração das especificidades de cada tipo de software e também da pouca consideração quanto à opinião dos usuários. Visando completar essas duas lacunas, propõe-se no presente trabalho o desenvolvimento de uma metodologia de avaliação ergonômica e de qualidade de software interativo baseada em técnicas de preferência declarada. Pretende-se, com essa ferramenta, sistematizar um conjunto de critérios a serem considerados na tarefa de avaliação desse tipo de produto.

A metodologia proposta tem uma estrutura hierárquica composta por dois módulos: no Módulo 1 são avaliadas as características básicas do software. Como características básicas consideram-se critérios ergonômicos e de qualidade que devem ser observados para qualquer tipo de software interativo. Para essa metodologia as características a serem analisadas baseiam-se nos critérios ergonômicos e de qualidade das normas NBR e ISO, estudados, respectivamente, nos capítulos 2 e 3 desse trabalho.

Caso, no Módulo 1, o software que está sendo analisado satisfaça certos requisitos, então a avaliação passa para o próximo estágio, o Módulo 2. Nesse módulo são avaliadas as características específicas do software, isto é, as peculiaridades de cada tipo de software interativo. Tais especificidades serão determinadas a partir de experimentos usando técnicas de preferência declarada, tal como vistas no Capítulo 4.

A estruturação de cada um desses módulos que compõem a metodologia de avaliação proposta é descrita a seguir.

## **5.2. Elaboração da Metodologia Proposta**

Para estruturar os módulos 1 e 2 da metodologia, foram seguidas quatro etapas, nomeadamente, as etapas de análise, concepção, valoração e revisão final. Cada uma das etapas está descrita na sequência.

### **5.2.1. Etapa de Análise**

Nessa primeira etapa da estruturação da metodologia, foram realizadas pesquisas bibliográficas sobre os assuntos pertinentes, com o objetivo de contextualizar o problema. Nesse caso, para a estruturação do Módulo 1, foram estudadas questões referentes à ergonomia e à qualidade de software. Como resultado dessas pesquisas, foi possível analisar os critérios e as perguntas utilizadas nos modelos de avaliação pesquisados.

Especial atenção foi dada aos Critérios Ergonômicos de Scapin e Bastien (1997), ao Diagnóstico de Software Interativo de Barthelet (1988), ao Método Iterativo de Concepção de Valentin, Vallery e Lucongsang (1993), à Norma NBR 13596 (1996) e à Norma ISO 9241 (1992). A lista de verificação que compõe este primeiro módulo da metodologia reflete a integração e condensação de recomendações existentes nesses métodos.

Durante essa etapa pôde-se constatar que a maior parte das metodologias de avaliação ergonômica e de qualidade não considera as especificidades de cada tipo de software, avaliando da mesma maneira, por exemplo, softwares educacionais e bancos de dados. Os resultados dessas pesquisas forneceram suporte técnico e metodológico para a elaboração do Módulo 2 da metodologia proposta, o qual incorpora as características específicas do software levando em consideração a opinião dos usuários. Para que isso fosse possível, foram realizados estudos referentes às técnicas de preferência declarada. O resultado desses estudos apresenta-se detalhado no capítulo 4 desse trabalho.

## 5.2.2. Etapa de Concepção

### 5.2.2.1. Etapa de Concepção do Módulo 1

Para a concepção do Módulo 1 de avaliação das características básicas de softwares interativos, inicialmente elaborou-se um *checklist* preliminar baseado nas metodologias estudadas na etapa anterior e apresentadas nos capítulos 2 e 3 desse trabalho. Um *checklist* nada mais é do que uma lista de verificação composta por perguntas que devem ser respondidas por um avaliador que, necessariamente, tenha tido algum contato direto com o objeto de avaliação (nesse caso, que tenha navegado pelo menos uma vez pelo software).

A partir de um teste com esse *checklist* preliminar e discussões com conhecedores da área, foram efetuadas correções e adaptações no mesmo, visando sempre torná-lo o mais claro e completo possível e, ao mesmo tempo, fazendo com que o mesmo não seja muito extenso.

O *checklist* para avaliação dos critérios básicos de um software é composto por 47 perguntas divididas em 8 classes. Cada classe está associada a um dos critérios considerados relevantes pela autora da tese para qualquer tipo de software interativo, quais sejam:

- Dispositivos de Apresentação e Linguagem de Interação, onde são avaliados o vocabulário, a legibilidade, a clareza e a homogeneidade das informações e comandos;
- Controle do Usuário, onde se avalia qual o nível de controle que o usuário tem sobre as ações e o seqüenciamento das mesmas;
- Adaptabilidade, onde é avaliado se o software é passível de adaptação de acordo com as especificidades, níveis de experiência ou preferências do usuário, tanto no que diz respeito a sua interface quanto ao diálogo;
- Gestão de Erros, onde são avaliadas a forma de proteção e correção de erros, bem como a qualidade das mensagens de erro;
- Carga de Trabalho, onde se avalia basicamente a brevidade de ações e a carga de memorização;
- Usabilidade, onde são avaliados critérios que identificam se o software é fácil de ser operado;
- Eficiência, esta avaliada em relação ao tempo e a quantidade de recursos utilizados; e
- Funcionalidade, onde se avalia se de fato o software realiza o que propõe e se essa realização ocorre de maneira adequada.

A listagem completa das perguntas associadas a cada um desses critérios será apresentada mais adiante, no sub-item 5.3.1 desse capítulo, sob o título Formulário de Avaliação.

É interessante ressaltar que o avaliador, para efetivação desse primeiro módulo, não necessariamente deve ter conhecimento da tarefa a ser realizada, devendo, porém, ter conhecimentos sobre ergonomia e qualidade de software, principalmente no que diz respeito ao vocabulário utilizado.

Concluindo, quanto à estrutura do módulo 1 da metodologia, cabe reforçar a idéia de que ela é fixa, isto é, as classes e perguntas serão sempre as mesmas, independente do tipo de software que está se querendo avaliar.

#### 5.2.2.2. Etapa de Concepção do Módulo 2

A questão principal quanto à estruturação desse módulo é o fato de poder existir um grande número de atributos do software que influenciam a preferência dos usuários. Sendo assim, quando isso acontece, o mais viável, em termos de planejamento, é a divisão desses em áreas de interesse, basicamente seguindo o que Louviere (1984) denominou de Integração Hierárquica da Informação. Nessa estratégia os atributos relevantes são divididos em conjuntos seguindo uma certa lógica ou característica comum evidente. Cada um dos conjuntos deve ser tratado separadamente, levando também em consideração o fato de que os atributos devem ser ortogonais, isto é, ter o mínimo de correlação entre si (SOUZA, 1999, p. 65). Assim, para cada área de interesse considerada, será necessário aplicar um experimento de preferência declarada, visando assim estabelecer o que será chamado de função utilidade local, isto é, a função utilidade relativa a cada área de interesse.

Para a efetiva aplicação do experimento e conseqüente obtenção da função utilidade local, é necessário primeiro descrever claramente cada um dos níveis dos atributos. Esses basicamente podem ser quantitativos (discretos ou contínuos) ou qualitativos, sendo os primeiros representados por um número ou um intervalo numérico e os últimos representados por expressões semânticas. A seguir devem ser identificadas todas as possíveis combinações entre os diferentes níveis dos atributos. Assim sendo, tem-se as chamadas alternativas.

Tendo conhecido, para cada área de interesse identificada, o número de atributos e seus respectivos níveis, é necessário então organizar como será feita a apresentação aos entrevistados das alternativas restantes. Para tanto, na maioria das vezes, faz-se necessário organizar as alternativas em blocos.

Finalmente, após a confecção dos cartões representando cada uma das alternativas, dá-se de fato a aplicação da pesquisa de preferência declarada para cada uma das áreas de interesse individualmente. Além disso, faz-se essencial também a aplicação de uma pesquisa de preferência declarada que identifique o grau de importância de cada uma das áreas de interesse consideradas, de forma a integrar as informações obtidas estatisticamente para cada área em separado (LOUVIERE e TIMMERMANS, 1990). Assim, ter-se-á condições de montar uma única função utilidade que represente a importância de cada um dos atributos considerados relevantes na avaliação do software em questão.

### 5.2.3. Etapa de Valoração

Uma vez definidos os critérios de avaliação e as respectivas perguntas, fez-se necessário definir também como seriam tratadas quantitativamente as informações obtidas com o *checklist* e com as entrevistas aos usuários, de forma que ao final de cada módulo fosse possível obter um parecer quanto à conformidade do software avaliado, segundo os critérios considerados.

#### 5.2.3.1. Etapa de Valoração do Módulo 1

Para a quantificação das questões presentes no Módulo 1, o avaliador é convidado a associar a cada questão um peso  $p_i$ , que representará o grau de importância daquela questão na avaliação. Desse modo, para a metodologia proposta, existem três graus de importância:

- Questões muito importantes, que recebem peso  $p_i = 1,5$ ;
- Questões importantes, que recebem peso  $p_i = 1$ ; e
- Questões onde o critério não se aplica, as quais recebem peso  $p_i = 0$ .

Cada questão tem apenas três alternativas de resposta e a cada uma delas é associado um valor numérico  $a_i$ , como segue:

- A resposta sim tem associado a ela o valor  $a_i = 1$ ;
- A resposta parcialmente tem associada a ela o valor  $a_i = 0,5$ ; e
- A resposta não tem associada a ela o valor  $a_i = 0$ .

Sendo assim, para o critério  $j$ , com um conjunto  $k$  de perguntas a serem respondidas, calcula-se um valor percentual, denotado  $VC_j$ , que representa a nota que o critério  $j$  alcançou segundo as respostas e os pesos dados pelo avaliador às perguntas feitas. Essa nota é calculada então conforme a expressão 5.1:

$$VC_j = \frac{\sum_{i=1}^k p_i a_i}{\sum_{i=1}^k p_i} \cdot 100 \quad (5.1)$$

Tendo feito isso para cada um dos critérios básicos, para a metodologia proposta devem ser verificados então os seguintes requisitos:

- Requisito 1) Taxa individual de aceitação

$$VC_j \geq 30 \quad \forall j = 1, 2, \dots, n$$

- Requisito 2) Taxa média de aceitação

$$\overline{VC_j} \geq 55, \text{ onde } \overline{VC_j} = \frac{\sum_{j=1}^n VC_j}{n} \text{ é a média de todos os critérios } VC_j.$$

O primeiro requisito visa garantir que o software tenha pelo menos 30% de conformidade com cada um dos critérios individualmente. Isso porque, caso ocorra o contrário, o software pode acabar não sendo útil para realizar a tarefa pretendida. Por exemplo, um software que obtém um valor de 20% para o atributo gestão de erros pode ser tão pouco confiável que acaba por inviabilizar sua adoção.

O segundo requisito visa garantir que a média aritmética de todos os critérios deve ser de, no mínimo, 55%. Isso quer dizer que, para satisfazer esse requisito, o software deve ter



uma taxa média de aceitação de pelo menos 55% de conformidade com os critérios básicos considerados.

Caso os dois requisitos sejam satisfeitos simultaneamente, deve-se então continuar a análise do software, partindo para a avaliação dos critérios específicos (módulo 2). Caso um dos requisitos falhe, faz-se necessária uma análise bastante cuidadosa do problema, cabendo ao avaliador, juntamente com as pessoas interessadas no diagnóstico do software, decidir quanto à continuidade ou não da avaliação.

#### 5.2.3.2. Etapa de Valoração do Módulo 2

A quantificação dos atributos pertencentes ao Módulo 2 se dá primeiramente através dos coeficientes  $\alpha_{ij}$  das funções utilidades que podem ser calibrados por um método de máxima verossimilhança ou de mínimos quadrados. Supõe-se, em nível de exemplificação, a seguinte situação:

- Foram identificadas 3 áreas de interesse.
- Na área  $A_1$  tem-se 5 atributos.
- Na área  $A_2$  tem-se 3 atributos.
- Na área  $A_3$  tem-se 4 atributos.

Desse modo, o software identifica três funções utilidade locais:

- $F(A_1) = \alpha_{11}x_{11} + \alpha_{12}x_{12} + \alpha_{13}x_{13} + \alpha_{14}x_{14} + \alpha_{15}x_{15}$
- $F(A_2) = \alpha_{21}x_{21} + \alpha_{22}x_{22} + \alpha_{23}x_{23}$
- $F(A_3) = \alpha_{31}x_{31} + \alpha_{32}x_{32} + \alpha_{33}x_{33} + \alpha_{34}x_{34}$

onde

$x_{ij}$  são os valores que serão associados a cada um dos atributos; e

$\alpha_{ij}$  são os coeficientes ponderadores, que devem ser calibrados segundo o procedimento estatístico da máxima verossimilhança.

Além disso, a partir do grau de relevância de cada área segundo o entrevistado, pode-se identificar os coeficientes ( $\beta_i$ ) que determinam na verdade o peso relativo de cada área de interesse, esses também calibrados pelos mesmos métodos descritos anteriormente. Assim,

obtem-se então uma função utilidade integradora, aqui chamada de função utilidade geral. Considerando o exemplo anterior, a função utilidade geral tem então a seguinte expressão:

$$FU = \beta_1 F(A_1) + \beta_2 F(A_2) + \beta_3 F(A_3)$$

Generalizando, a função utilidade geral, considerando-se  $n$  áreas de interesse, terá sempre a forma da expressão 5.2:

$$FU = \beta_1 F(A_1) + \beta_2 F(A_2) + \beta_3 F(A_3) + \dots + \beta_n F(A_n) \quad (5.2)$$

Partindo do pressuposto que todas as utilidades locais alcancem individualmente seus valores máximos, tem-se então identificada a função utilidade máxima, que irá equivaler, na metodologia em questão, a 100% de conformidade com os atributos especificados. Assim, quando um software específico for avaliado segundo esses atributos, basta calcular sua função utilidade geral de acordo com a expressão 5.2. De posse desse valor, basta aplicar uma regra de três simples para identificar o quão conforme o mesmo está em relação aos atributos considerados.

Por exemplo, supõe-se que a partir dos experimentos de preferência declarada obteve-se a seguinte função utilidade geral para análise de um software educacional:

$$FU = 0,34F(A_1) + 0,52F(A_2) + 0,49F(A_3) + 0,16F(A_4)$$

sendo que os valores máximos para as funções utilidade locais são, respectivamente:

- $F(A_1) = 1,6$
- $F(A_2) = 2,7$
- $F(A_3) = 0,86$
- $F(A_4) = 1,44$

Assim, o valor máximo da função utilidade é 2,5998, o que, nesse caso, equivale a 100% de conformidade com os atributos. Supondo agora que, avaliando especificamente o software educacional “ABC”, o mesmo obteve para a função utilidade geral o valor de 1,716. Dessa forma, pode-se dizer que o mesmo está 66% conforme aos atributos considerados relevantes para um software educacional.

#### **5.2.4. Etapa de Revisão Final**

Nessa etapa final de elaboração da metodologia, o Módulo 1 foi revisto por profissionais com conhecimentos sobre técnicas de avaliação de software. Sugestões para sua melhoria foram registradas para subsequente análise. Aquelas julgadas pertinentes foram implementadas e podem ser resumidas da seguinte maneira:

- Reformulação do texto das questões Q1.7, referente ao critério Dispositivos de Apresentação e Linguagem de Interação e Q3.2, referente ao critério Adaptabilidade.
- Alteração no formato de apresentação do Formulário de Avaliação, incluindo após cada questão espaço adequado para que o respondente insira tanto o peso quanto o valor da resposta dada a cada uma destas.

O formato de apresentação do Formulário de Avaliação será apresentado no próximo item deste capítulo.

No que se refere ao segundo módulo da metodologia proposta, devido ao processo bastante bem estruturado e aceito por pesquisadores que fazem uso das técnicas de preferência declarada, apenas foram realizadas conversas informais com profissionais com experiência na área. Nessas, o seqüenciamento de ações era relatado, e o pesquisador era então convidado a dar sua opinião sobre o desenrolar do experimento. Poucas considerações sobre a mesma foram feitas. Uma sugestão quanto à quantificação final da conformidade do software foi feita e essa, após melhorada, possibilitou a valoração final do software, tal como é apresentado atualmente na estrutura final da metodologia.

### **5.3. Estrutura da Metodologia**

#### **5.3.1. Estrutura do Módulo 1**

Através da Figura 5.1 pode-se visualizar a estrutura hierárquica da metodologia de avaliação de software interativo que se propõe. A descrição detalhada da estrutura dos dois módulos que a compõem será feita na seqüência.

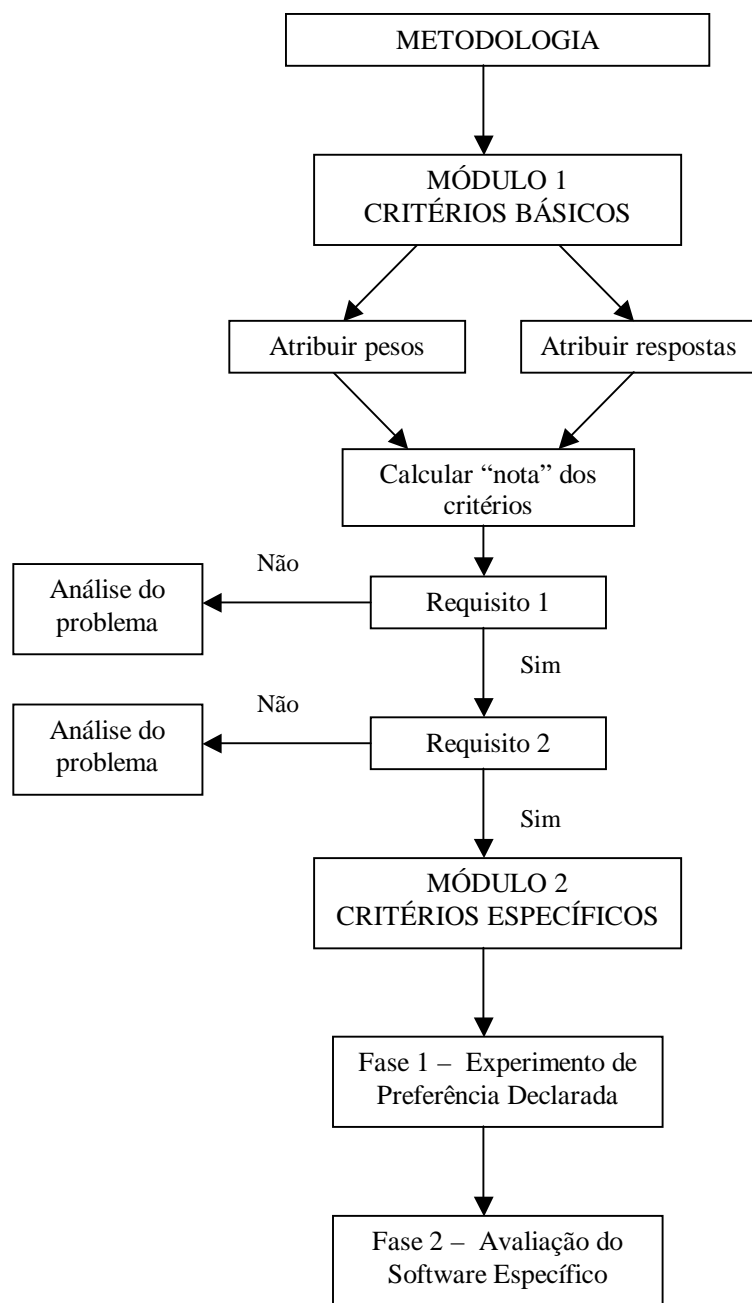


Figura 5.1 – Estrutura da metodologia proposta

Para descrever a estrutura da metodologia, será aqui apresentado, para a avaliação do Módulo 1, um Formulário de Avaliação, o qual deve ser preenchido pelo avaliador no momento da análise de um software específico. Como já mencionado, o Módulo 1 da metodologia é composto por 8 critérios básicos a serem avaliados através de 47 questões. A

cada uma das questões deve ser associado um peso de acordo com seu grau de importância, conforme explicitado no item 5.2.3.1 deste capítulo. Sugere-se que esta seja a primeira etapa a ser realizada no momento da avaliação do software.

Após serem identificados todos os pesos relativos a cada uma das 47 questões, o avaliador deve responder a cada uma dessas, sendo os valores atribuídos aqueles pré-definidos no item 5.2.3.1 deste capítulo.

De posse desses valores, faz-se necessário verificar a satisfação ou não dos requisitos identificados na metodologia, requisitos esses também explicitados no referido item. Sendo satisfeitos os requisitos, o avaliador deve passar então à avaliação dos critérios específicos do software. Conforme já mencionado nesse texto, caso um dos requisitos não seja satisfeito, é preciso analisar a causa desse problema para verificar se é possível não considerá-lo como uma restrição rígida. No caso, por exemplo, de se estar avaliando a performance de um protótipo de software, esse diagnóstico de “falha” em alguns critérios é uma informação deveras importante, servindo como *feedback* para a equipe de desenvolvimento do referido produto. Assim, nesse caso, deve-se continuar o processo de avaliação, visando a identificação de outros pontos problemáticos. Outra situação particular onde faz-se necessária uma análise mais detalhada do problema que gerou uma não conformidade é quando a média dos critérios praticamente atingiu o valor estipulado (Requisito 2), ficando esta, por exemplo, em 54%. Assim, cabe ao avaliador decidir se essa diferença realmente é significativa a ponto de desconsiderar um software do processo de avaliação.

O Formulário de Avaliação, assim como é apresentado aos avaliadores, tem a estrutura como apresentada na seqüência.

### **Formulário de Avaliação**

Identifique a relevância de cada questão de acordo com o tipo de produto que se deseja avaliar. Para tanto, associe um peso  $p_i$  a cada uma delas de acordo com a legenda:

$p_i = 1,5$  – questão muito importante

$p_i = 1,0$  – questão importante

$p_i = 0$  – questão não se aplica

Responda as questões a seguir conforme indica a legenda:

S = 1,0 – sim

P = 0,5 – praticamente

N = 0 – não

### **Critério 1. Dispositivos de Apresentação e Linguagem de Interação**

- Q1.1) O vocabulário usado é familiar ao usuário e está de acordo com a tarefa a ser realizada?  
**Peso ( ) Resposta ( )**
- Q1.2) O vocabulário e as abreviações usadas são facilmente compreensíveis e, se necessário, de fácil memorização?  
**Peso ( ) Resposta ( )**
- Q1.3) Um comando, botão ou opção de menu, quando selecionado, realiza sempre a mesma ação?  
**Peso ( ) Resposta ( )**
- Q1.4) As informações na tela são legíveis, padronizadas e bem distribuídas espacialmente?  
**Peso ( ) Resposta ( )**
- Q1.5) O tamanho da fonte e cores são usados adequadamente?  
**Peso ( ) Resposta ( )**
- Q1.6) As informações da mesma natureza (entrada de dados, botões, textos e títulos) aparecem sempre no mesmo local e seguem o mesmo padrão em todas as telas?  
**Peso ( ) Resposta ( )**
- Q1.7) Todas as entradas de dados feitas pelo usuário são mostradas na tela, mesmo que através de símbolos como, por exemplo “\*”?  
**Peso ( ) Resposta ( )**
- Q1.8) O software fornece informações sobre o estado do processamento das informações quando esse é muito longo?  
**Peso ( ) Resposta ( )**
- Q1.9) É possível distinguir visualmente, através de pastas, por exemplo, áreas que tenham diferentes funções?  
**Peso ( ) Resposta ( )**

**Critério 2. Controle do Usuário**

- Q2.1) O usuário precisa sempre confirmar, através da tecla ENTER, a inicialização do processamento das informações e a efetivação das entradas de comandos?

**Peso ( ) Resposta ( )**

- Q2.2) O usuário pode cancelar uma ação sempre que desejar, voltando assim à situação anterior?

**Peso ( ) Resposta ( )**

- Q2.3) O usuário pode interromper uma tarefa e retomá-la posteriormente do mesmo ponto?

**Peso ( ) Resposta ( )**

- Q2.4) O usuário pode controlar a velocidade de interação, principalmente no que diz respeito à entrada de dados?

**Peso ( ) Resposta ( )**

- Q2.5) O usuário pode controlar o que aparece na tela, desde o posicionamento do cursor até a quantidade de informações apresentadas?

**Peso ( ) Resposta ( )**

- Q2.6) O usuário pode ter acesso a todas as informações a qualquer momento?

**Peso ( ) Resposta ( )**

**Critério 3. Adaptabilidade**

- Q3.1) O software permite que o usuário configure a tela e os dispositivos de entrada e saída de dados de acordo com suas preferências e necessidades?

**Peso ( ) Resposta ( )**

- Q3.2) O software disponibiliza diferentes tipos de diálogo para diferentes níveis de habilidade e experiência do usuário (principalmente diferenciando iniciantes e experientes)?

**Peso ( ) Resposta ( )**

- Q3.3) O software prevê navegação passo-a-passo para iniciantes e teclas de atalho para os mais experientes?

**Peso ( ) Resposta ( )**

- Q3.4) O software dispõe informações diferenciadas, inclusive sobre erros ocorridos, de acordo com o nível de conhecimento do indivíduo?

**Peso ( ) Resposta ( )**

- Q3.5) O software permite que o usuário inclua comandos personalizados e configure parâmetros de tempo operacional de acordo com suas necessidades individuais?

**Peso ( ) Resposta ( )**

#### **Critério 4. Gestão de Erros**

- Q4.1) O software solicita a confirmação de uma ação sempre que essa tenha sérias conseqüências, como, por exemplo, a perda de dados ou o não processamento de alguma informação?

**Peso ( ) Resposta ( )**

- Q4.2) O software protege funções e comandos perigosos?

**Peso ( ) Resposta ( )**

- Q4.3) O software nunca ou quase nunca apresenta falhas?

**Peso ( ) Resposta ( )**

- Q4.4) O software é capaz de recuperar dados no caso de ocorrência de falhas?

**Peso ( ) Resposta ( )**

- Q4.5) O software apresenta uma mensagem explicativa, indicando a opção correta, quando o usuário seleciona uma função inválida?

**Peso ( ) Resposta ( )**

- Q4.6) As mensagens de erro são claras, instruindo o usuário para a correção do problema?

**Peso ( ) Resposta ( )**

- Q4.7) O software, quando solicitado, fornece explicações adicionais durante a correção de erros?

**Peso ( ) Resposta ( )**

- Q4.8) O software permite que o usuário corrija os erros direta e imediatamente, sem ter que mudar o estado atual do sistema?

**Peso ( ) Resposta ( )**

- Q4.9) O software permite que o usuário corrija somente os comandos ou a parte digitada incorretamente?

**Peso ( ) Resposta ( )**



**Critério 5. Carga de Trabalho**

- Q5.1) Dados que podem ser derivados de outros já acessíveis ao sistema são obtidos automaticamente?  
**Peso ( ) Resposta ( )**
- Q5.2) O software pode importar dados de uma tela para outra, evitando assim a memorização do usuário?  
**Peso ( ) Resposta ( )**
- Q5.3) O software disponibiliza somente as informações necessárias para aquele momento?  
**Peso ( ) Resposta ( )**
- Q5.4) O software possibilita o acesso direto a um ponto específico do mesmo, evitando assim toda uma navegação até esse?  
**Peso ( ) Resposta ( )**
- Q5.5) O software exibe valores *default* apropriados?  
**Peso ( ) Resposta ( )**
- Q5.6) O software, quando possível, minimiza os passos na execução de uma tarefa?  
**Peso ( ) Resposta ( )**

**Critério 6. Usabilidade**

- Q6.1) É fácil aprender a usar o software?  
**Peso ( ) Resposta ( )**
- Q6.2) Existem ferramentas como, por exemplo, tutoriais interativos, que auxiliam o aprendizado do software?  
**Peso ( ) Resposta ( )**
- Q6.3) É fácil operar o software?  
**Peso ( ) Resposta ( )**
- Q6.4) O software dispõe de orientação e ajuda para sua operação sempre que necessário?  
**Peso ( ) Resposta ( )**
- Q6.5) O software dispõe de um glossário para auxiliar o usuário na compreensão de termos técnicos?  
**Peso ( ) Resposta ( )**

### **Critério 7. Eficiência**

- Q7.1) O tempo de resposta do software é aceitável, considerando o “tamanho” da tarefa?

**Peso ( ) Resposta ( )**

- Q7.2) Considerando a quantidade de recursos utilizados pelo software (impressora, outros produtos de software, manutenção e suporte), pode-se dizer que o mesmo é eficiente?

**Peso ( ) Resposta ( )**

### **Critério 8. Funcionalidade**

- Q8.1) O software se propõe a realizar a tarefa requerida?

**Peso ( ) Resposta ( )**

- Q8.2) O software faz o que propõe corretamente, disponibilizando as funções adequadas para a realização da tarefa?

**Peso ( ) Resposta ( )**

- Q8.3) O software posiciona as funções nos menus respeitando o modo de trabalho dos usuários, isto é, colocando as opções mais utilizadas estrategicamente?

**Peso ( ) Resposta ( )**

- Q8.4) O software interage com outros sistemas essenciais na realização da tarefa?

**Peso ( ) Resposta ( )**

- Q8.5) O software está de acordo com normas, leis e padrões da área na qual se insere a aplicação?

**Peso ( ) Resposta ( )**

### **5.3.2. Estrutura do Módulo 2**

Para analisar as características específicas de um software, o trabalho é bem mais minucioso, sendo necessário identificar quais os critérios relevantes a serem analisados e seus respectivos pesos. Somente após isso é que a efetiva avaliação poderá ser feita, tendo como resposta final um índice de conformidade do software em relação aos critérios considerados.

Dessa forma, esse módulo de avaliação foi estruturado em duas fases distintas porém, hierarquicamente dependentes, conforme já apresentado na Figura 5.1: a Fase 1 – Experimento de Preferência Declarada e a Fase 2 – Avaliação do Software Específico.

Para realização da Fase 1, faz-se necessária a contratação de pessoal especializado no desenvolvimento e realização de experimentos de preferência declarada. Essa equipe, para atingir os objetivos pretendidos na metodologia proposta, deve seguir uma lista de procedimentos que consiste nos seguintes passos:

- Definir uma lista preliminar de atributos;
- Dividir os atributos, se necessário, em áreas de interesse;
- Se não houver divisão em áreas, desenhar um único experimento de preferência declarada;
- Se houver mais de uma área de interesse, para cada uma dessas, desenhar um experimento de preferência declarada. Além disso, desenhar um outro experimento que identifique a importância relativa de cada uma dessas áreas;
- Identificar um valor numérico para ser possível quantificar cada nível de cada atributo qualitativo, em geral usando valores inteiros como -1, 0 e 1;
- Aplicar o(s) experimento(s) de preferência declarada, entrevistando uma amostra significativa e representativa do público-alvo do software;
- Encontrar os coeficientes de calibração do modelo, (podendo ser usado o software LMPC);
- Se não houver divisão em áreas, montar a função utilidade geral;
- Se houver mais de uma área de interesse, montar a função utilidade para cada uma das áreas de interesse, montando, posteriormente, a função utilidade geral; e
- Identificar a função utilidade geral máxima que pode ser alcançada, o que equivalerá a 100% de conformidade.

De posse desses valores, a equipe especialista pode ser dispensada, ficando a continuidade dos trabalhos – Fase 2 – sob responsabilidade do avaliador.

Assim sendo, na segunda fase de avaliação dos critérios específicos, o avaliador está de posse dos pesos dos atributos e conseqüentemente das funções utilidade (para cada área de interesse, geral e geral máxima), como também da valoração dos níveis de cada atributo. Parte-se então para a análise particular de um software específico.

Neste momento surge a necessidade de se identificar a existência de algum atributo que seja expressamente indispensável para o tratamento do problema a partir do software,

para aquele usuário específico. Caso não haja um atributo deste tipo, segue-se com a avaliação normalmente. Caso exista algum atributo deste tipo, então há a necessidade de se fazer uma análise mais detalhada do problema (não havendo possibilidade então de considerar para análise softwares que não contemplem esse atributo).

Desse modo, o que se está analisando nesse momento são as características de um software em particular segundo as expectativas dos usuários em potencial, essas identificadas a partir dos atributos levantados na fase anterior. Para cada atributo considerado nas funções utilidade, deve-se atribuir o valor numérico correspondente ao nível em que o software avaliado se encaixa. Tendo sido feito isso em todas as funções utilidade das diferentes áreas de interesse, resta calcular, para aquele software em especial, a função utilidade geral. De posse desse valor, basta somente calcular, por uma regra de três simples com o valor da função utilidade geral máxima (obtido na Fase 1), qual o percentual de conformidade desse software.

## **CAPÍTULO 6 - APLICAÇÃO DA METODOLOGIA DE AVALIAÇÃO**

### **6.1. Introdução**

A aplicabilidade da metodologia proposta nesse trabalho é considerada tomando como estudo de caso os softwares para roteirização de veículos, amplamente utilizados por empresas que fazem distribuição de produtos, como, por exemplo, os operadores logísticos. Para tanto, é primordial fazer algumas considerações iniciais sobre essa área de estudos, para só então ser possível detalhar o projeto experimental e os resultados obtidos para os coeficientes de ajuste de escolha dos atributos mais relevantes, utilizando para tanto as técnicas de preferência declarada. Vale ressaltar que como o módulo 1 da metodologia é fixo, só serão detalhados aqui o desenvolvimento e a aplicação do módulo 2 da mesma no contexto dos softwares para roteirização de veículos.

### **6.2. O Contexto da Aplicação**

O transporte, juntamente com a manutenção de estoques e o processamento de pedidos, é considerado como atividade primária da logística devido a sua importância para o alcance dos objetivos logísticos, ou seja, menor custo e maior nível de serviço. Além disso, as atividades de transporte são em geral responsáveis por elevados custos, absorvendo, em média, de um a dois terços dos custos logísticos (BALLOU, 2001).

As atividades de gerenciamento e operação de transporte são definidas basicamente pela economia de escala e pela economia de distância. Isso quer dizer que serão adotadas aquelas diretrizes de transporte que acarretam a diminuição do custo de transporte por unidade de peso e por unidade de distância, atendendo ao mesmo tempo as expectativas de serviço dos clientes (BOWERSOX e CLOSS, 2001, p. 280). Sendo assim, para que o processo de distribuição seja realizado eficientemente, a empresa deve abordar o planejamento e a execução de suas atividades de transporte de maneira racional e tendo uma visão total das economias envolvidas (BODIN *et al*, 1983, p. 69).

As atividades de transporte, assim como as demais atividades logísticas (armazenagem, processamento de pedidos, aquisição etc) estão divididas em três grandes níveis de decisão: estratégico, tático e operacional. Decisões estratégicas são aquelas que têm reflexo em longo prazo, como, por exemplo, a decisão quanto à localização de um depósito ou a seleção do modal de transporte. As decisões táticas são aquelas tomadas em médio prazo e em geral delegadas pelos gerentes da empresa. Questões quanto à definição de rotas e a decisão quanto ao uso sazonal de veículos são consideradas nesse nível. Por fim, as decisões operacionais são aquelas que afetam diretamente aqueles que efetivamente trabalham nas tarefas de distribuição, sendo que os reflexos quanto às decisões aparecem em curto prazo, em geral refletindo nas atividades diárias da empresa. Neste nível são tomadas decisões quanto a roteirização de veículos, agendamento e despacho de produtos. Para conseguir realizar essas últimas tarefas – roteirizar e programar – de forma a atingir os objetivos da empresa ao mínimo custo, é necessário um algoritmo de busca para identificar a melhor configuração das rotas e horários.

O problema de roteirizar e programar veículos é bastante complexo. É um processo que visa determinar um ou mais roteiros ou seqüências de paradas a serem efetuados por veículos de uma frota, de forma a “visitar um conjunto de pontos geograficamente dispersos, em locais pré-determinados, que necessitam de atendimento” (CUNHA, 2003). Ronen (1988, p. 137) esclarece que o termo roteirização (*routing*) refere-se ao planejamento da rota que cada veículo irá tomar, enquanto que o termo programação (*scheduling*) refere-se ao planejamento do horário em que cada rota deverá ser percorrida. Sendo assim, o problema de roteirizar veículos, em geral, envolve “o planejamento de um conjunto de rotas de mínimo custo, que inicia e termina em pelo menos um depósito central, através de uma frota de veículos que atende a um conjunto de clientes com demanda conhecida” (PELIZARO e DA SILVA, 2003, p.183).

Laporte (1992, p. 345) dá uma definição mais completa, considerando a roteirização de veículos como um problema a ser resolvido através do desenho da rota ótima de entrega ou coleta de um ou vários depósitos para um número de cidades ou clientes geograficamente distribuídos, sujeito a certas restrições. Nessa mesma linha de pensamento, para Partyka e Hall (2000), um problema real de roteirização é definido por três fatores fundamentais: decisões, objetivos e restrições. É necessário tomar decisões quanto a alocação de um grupo de clientes que devem ser visitados por um conjunto de veículos e respectivos motoristas,

identificando também o seqüenciamento das visitas. Objetiva-se, com a adoção de um roteiro otimizado, propiciar um serviço de alto nível aos clientes, mantendo os custos operacionais e de capital tão baixos quanto possível. Contudo, há certas restrições reais que devem ser consideradas, como, por exemplo, os limites de tempo nas jornadas de trabalho e as restrições de trânsito (velocidade, horário de circulação, carga e descarga e tamanho máximo do veículo).

Para tratar diariamente essa gama de variáveis, têm sido desenvolvidas ferramentas computacionais que tem como objetivo, a partir de dados como localização de clientes, tempos gastos com as entregas e velocidade média dos veículos, encontrar as rotas a serem seguidas para efetivar a distribuição aos clientes, respeitando as restrições existentes e visando sempre a minimização dos custos.

### **6.2.1. Softwares de Roteirização**

#### 6.2.1.1. Evolução

Os primeiros sistemas para roteirização e programação de veículos operavam em *mainframes* e, portanto era necessário muita paciência no aguardo das respostas devido ao elevado tempo de processamento. Além disso, não existiam recursos gráficos e não havia possibilidade de testar alterações feitas manualmente, o que acabava por prejudicar o entendimento e aceitação das soluções. Posteriormente surgiram os sistemas auxiliados por computador que, apesar de um pouco mais rápidos, não forneciam uma solução ótima. Cabia ao usuário examinar as diferentes alternativas fornecidas pelo sistema para então selecionar a que ele julgasse mais adequada.

Mais adiante, em meados dos anos 80, esses sistemas computacionais foram evoluindo e passaram a considerar um número maior de restrições reais, com alguns, inclusive, já disponibilizando recursos gráficos. Nesta época surgiram vários sistemas com essa finalidade e assim seu uso foi se proliferando e conseqüentemente seu custo foi diminuindo (GOLDEN, BODIN e GOODWIN, 1986).

A partir da década de 90, com o *boom* tecnológico, o advento da internet e do GPS (*Global Positioning System*) e com pesquisas mais avançadas sobre pesquisa operacional, os softwares de roteirização (ou roteirizadores) ficaram cada vez mais robustos, mais simples em

termos de interface e mais acessíveis às empresas. Além disso, as pressões dos clientes com relação a prazos, datas, horários e frequência de atendimento, o agravamento dos problemas de trânsito nos centros urbanos, o aumento da competição pelo mercado e a busca por eficiência fizeram aumentar o interesse e a demanda pelos softwares de roteirização disponíveis (CUNHA, 2003). No Brasil isso também ocorreu, porém de maneira menos grandiosa, principalmente após a estabilização da economia.

Atualmente, de acordo com Melo e Ferreira Filho (2001, p. 225), a maior parte dos roteirizadores disponíveis no mercado são do tipo SADE (Sistemas de Apoio à Decisão Espacial), ou seja, faz uso da tecnologia de Sistemas de Informação Geográfica (GIS). Essa tecnologia permite visualizar rotas e paradas em mapas, identificar e geocodificar todos os pontos de atendimento a partir do endereço dos clientes, além de armazenar qualquer informação referente aos mesmos. Dessa forma, os sistemas atualmente disponíveis são capazes de lidar com restrições mais complexas, são em geral mais flexíveis e robustos e, quanto à interface, apresentam melhores recursos gráficos e são mais fáceis de se aprender a manipular.

Em pesquisa realizada junto a empresas que comercializavam roteirizadores, Partyka e Hall em 1997 concluíram que ainda não havia no mercado um produto que integrasse otimização de rotas, navegação e rastreamento de veículos. Previam, para um futuro não tão distante, softwares que integrassem roteirização, rastreamento de frota e carga, podendo também estar este integrado à programação da produção. Viam assim um software logístico totalmente integrado, capaz de gerenciar o fluxo de material do suprimento até o cliente (PARTYKA e HALL, 1997). Os mesmos autores, numa segunda pesquisa realizada três anos após, identificaram uma mudança significativa na indústria por causa de grandes avanços tecnológicos. Previam com isso, para os anos seguintes que, através de uma plataforma flexível e portátil, os motoristas pudessem acessar informações em tempo real sobre o tráfego e o tempo, trocar mensagens com os clientes, atualizar relatórios de inventário e requisitar assistência caso necessário (HALL e PARTYKA, 2000). Na última pesquisa realizada, a maior parte dessas previsões já havia se concretizado, sendo que alguns softwares de roteirização disponíveis no mercado estavam fazendo uso de comunicação de dados sem fio e sendo desenvolvidos inclusive a partir de algoritmos genéticos. A maior novidade da pesquisa foi a constatação de que algumas empresas que desenvolvem roteirizadores estão começando a criar bibliotecas para disponibilizar os códigos de roteirização utilizados em seus softwares.



Com isso, outros desenvolvedores podem fazer uso desses e simplesmente customizar seus produtos de forma a melhor atender as exigências dos clientes (HALL, 2002).

Empresas que fazem constantes cotações de preço, por terem grande frequência de transporte para vários destinos e com volumes significativos, são o público-alvo dos softwares de roteirização de veículos. Entre os maiores compradores estão os atacadistas, as redes de supermercado e as indústrias, sendo adquiridos também por transportadoras com frotas reduzidas de 10, 12 ou 15 veículos (REVISTA TECNOLÓGICA, 2003).

#### 6.2.1.2. Estrutura de um Software de Roteirização

Softwares de roteirização, ou roteirizadores, são ferramentas tecnológicas que em geral melhoram a qualidade e produtividade do processo de distribuição. Geralmente instalados no sistema de vendas, são de fato

sistemas computacionais que, através de algoritmos, geralmente heurísticos, e uma apropriada base de dados, são capazes de obter soluções para problemas de roteirização e programação de veículos com resultados relativamente satisfatórios, consumindo tempo e esforço de processamento relativamente pequenos (MELO e FERREIRA F<sup>o</sup>, 2001, p. 224).

Os roteirizadores têm como objetivos principais:

- Maximizar a utilização da frota; e
- Reduzir custos com a distribuição, já que pretendem minimizar os tempos de trabalho e quilometragem.

Eles podem ser uma poderosa ferramenta de análise e simulação de estratégias de distribuição, contribuindo assim para a eficiência de toda a cadeia de suprimentos. Além disso, podem ser integrados a sistemas WMS (*Warehouse Management Systems*), direcionando a separação e o carregamento dos veículos, e a rastreadores, aumentando assim a segurança da carga transportada e o nível de serviço prestado ao cliente.

Os roteirizadores necessitam de dados iniciais como, por exemplo:

- A localização geográfica dos clientes, que pode ser inserida por um sistema de coordenadas ou a partir de um mapa digitalizado;
- Os tempos (médios) de espera para descarga do veículo;

- Os tempos gastos em trânsito;
- Os horários possíveis de recebimento dos clientes, também conhecidos como janelas de tempo;
- Os dias possíveis de recebimento;
- As restrições, se existirem, das áreas de circulação;
- A jornada de trabalho dos motoristas e auxiliares;
- O tipo e a capacidade dos veículos; e
- As características dos produtos a serem distribuídos.

Hall e Partyka alertam que esse tipo de software também deve ser capaz de lidar com um misto de recolhimento e entrega de cargas, o que pode complicar os cálculos da capacidade disponível do veículo; com diferentes especializações dos motoristas, já que pode ser interessante que os mesmos saibam reparar um equipamento ou dirigir diferentes tipos de veículo; e com a existência de múltiplas facilidades, como depósitos, centros de distribuição e plantas industriais (HALL e PARTYKA, 2000). Além disso, questões quanto à aleatoriedade da demanda e a dinamicidade da roteirização são questões que preocupam os administradores de frotas, e por isso também devem ser contempladas pelos softwares de roteirização (PARTYKA e HALL, 1997).

Após o processamento das informações, tarefa que em geral não despende muito tempo, esses softwares fornecem, em geral, como resultado:

- O roteiro e a programação de cada veículo, preferencialmente com recursos gráficos;
- O relatório de utilização do veículo, incluindo o relatório para consolidação de cargas e a ordem para o carregamento do veículo; e
- As notas fiscais já na seqüência das entregas.

O esforço computacional para o tratamento de todas as informações relevantes para a roteirização de veículos cresce exponencialmente com o tamanho do problema, ou seja, é um problema classificado como NP-difícil (*NP-hard*). Cunha (2003) argumenta que

Em termos práticos (...) não é possível resolver até a otimalidade problemas reais pertencentes à classe NP-difícil. Conseqüentemente, os métodos de solução de todos os softwares e aplicativos comerciais encontrados no

mercado para roteirização de veículos são heurísticos, isto é, não asseguram a obtenção da solução ótima do ponto de vista matemático.

A aquisição de um software por uma empresa, independente de sua funcionalidade, é válida se a relação custo-benefício tiver resultados positivos, ou seja, se a empresa for utilizá-lo constantemente, ele se pagará em pouco tempo de uso; porém, se a empresa estiver adquirindo-o somente para a conferência de uma prática já estabelecida e satisfatória ou para o uso esporádico, sua aquisição poderá ser pouco vantajosa.

Decidido quanto à adoção do software de roteirização, o problema agora está na escolha do produto a adotar. Para que seja tomada a decisão mais acertada é preciso analisar quais os requisitos que um software para essa finalidade deve disponibilizar.

#### 6.2.1.3. Requisitos de um Roteirizador

A partir do estudo de diferentes autores (GOLDEN, BODIN e GOODWIN, 1986; ASSAD, 1988; RONEN, 1988; BODIN, 1990; MELO e FERREIRA Fº, 2001; HALL, 2002), foram levantados os principais requisitos que os softwares de roteirização de veículo devem contemplar. Muitos dos requisitos são unanimidade entre os autores, como, por exemplo, a necessidade de o software considerar janelas de tempo; porém, outros requisitos são mais enfatizados por certos autores específicos. A listagem a seguir nada mais é do que a junção dos diferentes requisitos mencionados pelos autores. Assim sendo, baseado nesse levantamento, um bom roteirizador deve ser capaz de:

- Considerar uma ou múltiplas bases;
- Considerar diferentes tipos de veículos;
- Considerar janelas de tempo para cada cliente;
- Considerar tempos de carga e descarga;
- Considerar velocidades variáveis;
- Considerar a contratação de terceiros;
- Considerar os limites de capacidade (peso e volume) do veículo;
- Considerar múltiplos compartimentos por veículo;
- Considerar a duração máxima de um roteiro;
- Contabilizar as horas-extras dos motoristas;
- Considerar os horários de início e término da viagem;
- Considerar roteiros com pernoite e/ou troca de motorista;

- Considerar a jornada de trabalho do motorista;
- Considerar locais e/ou horários de parada fixos, por exemplo, para almoço;
- Considerar restrições de tamanho do veículo para um cliente específico;
- Inserir barreiras e restrições de circulação de veículos;
- Considerar mais de um roteiro por veículo;
- Considerar a possibilidade de coletas de retorno (*backhaul*);
- Inserir e tratar informações reais de tráfego;
- Possibilitar a mudança manual das soluções;
- Disponibilizar recursos gráficos, mapas principalmente;
- Geocodificar os pontos de paradas pelo endereço;
- Calcular valores de distância e tempo reais;
- Inferir sobre informações reais de tráfego;
- Emitir relatório de planejamento do carregamento;
- Emitir relatório de programação turno a turno ao motorista;
- Resolver problemas de tamanho ilimitado (sem restrições para o número de veículos e paradas, por exemplo);
- Roteirizar por arcos e por nós;
- Roteirizar em tempo real; e
- Interagir com outros softwares (rastreadores, por exemplo).

Com o software correto para roteirização de veículos, a empresa pode conseguir ganhos financeiros e melhora na qualidade dos serviços oferecidos, já que a partir do uso desse é comum obter-se resultados significativos, como o aumento do número de clientes atendidos, a redução da frota, a diminuição do custo operacional, a redução significativa dos tempos de entrega e das horas-extras dos motoristas etc (MELO e FERREIRA Fº, 2001, p. 230). O problema está em como selecionar um roteirizador dentre os disponíveis de forma que a empresa consiga alcançar esses resultados.

Novaes (2001, p. 312) argumenta que para se selecionar um software de roteirização, o ideal é que sejam definidas uma ou mais situações para testar os sistemas disponíveis no mercado antes de sua aquisição. Contudo, nem sempre é possível simular situações realistas para testes e, além disso, em alguns casos, não existe a possibilidade de testar o software antes

de adotá-lo. Para que esse processo de seleção seja realizado cautelosamente, evitando assim gastos desnecessários ou a escolha de uma ferramenta ineficiente, o referido autor sugere:

- Fazer uma análise quanto à necessidade de participação de um ou mais consultores na instalação e adaptação do software ao caso particular da empresa, já que isso vai ter influência direta nos prazos e custos;
- Verificar se o software tem restrições ou faz simplificações que podem de alguma forma influenciar a operacionalização ou a precisão dos resultados finais;
- Caso o software faça uso da representação digital da rede viária, verificar se a empresa fornecedora do software tem a mesma para disponibilização, pois, caso contrário, o desenvolvimento da mesma deverá ser feito pela empresa usuária, o que toma tempo e gera custos; e
- Caso o problema a ser resolvido tenha pontos de entrega variando diariamente, identificar se o software consegue lidar com essa variabilidade, pois se isso não for possível, o software não tem utilidade nessa situação.

Para Hall (2002), antes de se iniciar o processo de seleção de roteirizadores, devem ser feitas algumas considerações quanto a detalhes técnicos referentes ao problema a ser resolvido. O autor sugere que o responsável pelo gerenciamento de frotas responda às perguntas-chave:

- Qual o tamanho do problema a ser resolvido, medido em veículos, paradas e terminais?
- Quão frequentemente a solução será atualizada?
- Quão rapidamente o software deverá gerar a solução?
- Quem irá usar o software?
- Como a informação pode ser melhor apresentada ao usuário do software?
- Os usuários do software estão distribuídos em várias localidades?
- Quem irá instalar e fazer a manutenção do software?
- Quais sistemas de software devem interagir com o roteirizador?

Para a efetiva aplicação do módulo 2 da metodologia proposta, é necessário definir os atributos a serem considerados e o delineamento experimental da pesquisa de preferência declarada realizada para encontrar os pesos relativos a tais atributos.

### **6.3. O Projeto Experimental**

De acordo com a metodologia proposta, far-se-ia necessário realizar uma pesquisa que pudesse identificar quais os atributos de softwares para roteirização de veículos são considerados mais importantes na hora da adoção ou mesmo do planejamento para o desenvolvimento de um produto com essa finalidade. Como pôde-se observar anteriormente neste texto (item 6.2.1.3), há uma lista longa de atributos que caracterizam e podem diferenciar um produto de outro. Trabalhar de uma única vez com todos esses atributos seria praticamente impossível, principalmente devido ao elevado número de combinações que estes poderiam gerar.

A primeira etapa do projeto experimental foi então uma pesquisa prévia com estudiosos da área, sendo que essa pretendia identificar um número mais restrito de atributos para se trabalhar. Essa pesquisa se deu através da aplicação de um questionário, instrumento este que pode ser visto no Anexo 4. A partir da análise dos dados obtidos nesta primeira etapa, foram relacionados 16 dentre os 30 atributos como sendo os mais relevantes. Como 16 é ainda um número muito alto e sabendo do perigo que se incorre ao se apresentar aos entrevistados tarefas muito extensas e complicadas, seguindo a metodologia proposta, optou-se por subdividir esse conjunto maior em áreas de interesse, aqui nomeadas de conjuntos. De acordo com as características desses atributos selecionados, pôde-se criar 3 conjuntos, cujos delineamentos experimentais são apresentados na seqüência. Da mesma forma faz-se necessário detalhar o delineamento experimental geral que identifique por fim a expressão matemática da função utilidade global.

#### **6.3.1. Delineamento Experimental do Conjunto 1**

O Conjunto 1, denominado Quanto à Capacidade Algorítmica do Roteirizador, é formado pelos seguintes atributos:

- Janelas de tempo: horário pré-determinado por um cliente para o recebimento de mercadorias.
- Tempo de carga e descarga: o tempo que se irá levar para carregar e/ou descarregar o veículo.

- Considera a velocidade: um software pode trabalhar com uma velocidade fixa (por exemplo, a velocidade média) ou variável (considerando diferentes velocidades, dependendo da rodovia ou do horário, por exemplo).
- Horário de início e fim da viagem: a política de jornada de trabalho da empresa pode ser um fator importante a se considerar. Caso não haja nenhuma restrição na empresa de horário para início ou fim de uma viagem, então esse fator não é relevante.
- Roteiros por veículo: número de roteiros que o software pode alocar para cada veículo por dia.
- Restrições de circulação de veículos: horários onde é proibida a circulação de veículos de grande porte, rodízio de veículos de acordo com o número da placa e vias de mão única são exemplos de restrições de circulação de veículos.

A especificação de cada nível desses atributos, bem como sua valoração e a variável associada são mostrados na tabela 6.1. Pode-se notar que, para cada um dos atributos foram definidos 2 níveis: o nível associado ao valor zero, representando a forma mais básica do atributo ou mesmo a inexistência dele no software; e o nível associado ao valor 1, representando a forma mais completa do atributo, ou tão somente a sua existência ou consideração. Esse padrão irá se repetir também para os demais conjuntos constantes nesse experimento.

Tabela 6.1 – Níveis dos atributos do Conjunto 1

<b>Atributo</b>	<b>Valoração</b>	<b>Níveis</b>	<b>Variável</b>
Janelas de Tempo	0	Rígidas	$X_{11}$
	1	Flexíveis	
Tempo de carga e descarga	0	Não considera	$X_{12}$
	1	Considera	
Considera a velocidade	0	Fixa	$X_{13}$
	1	Variável	
Horário de início e fim da viagem	0	Não considera	$X_{14}$
	1	Considera	
Roteiros por veículo	0	Um único	$X_{15}$
	1	Múltiplos	
Restrições de circulação de veículos	0	Não considera	$X_{16}$
	1	Considera	

Deste modo, a função utilidade local referente ao Conjunto 1 fica sendo:

$$F(C1) = \alpha_{11}X_{11} + \alpha_{12}X_{12} + \alpha_{13}X_{13} + \alpha_{14}X_{14} + \alpha_{15}X_{15} + \alpha_{16}X_{16} \quad (6.1)$$

O delineamento experimental utilizado constitui-se em um ensaio fatorial fracionário conforme consta na tabela 6.2, onde em cada coluna são atribuídos os respectivos níveis dos atributos. Assim, cada alternativa fica então definida segundo as configurações dos níveis dos atributos.

Tabela 6.2 – Alternativas e níveis dos atributos do Conjunto 1

	<b>Atributos</b>					
	$X_{11}$	$X_{12}$	$X_{13}$	$X_{14}$	$X_{15}$	$X_{16}$
<b>A</b>	0	0	0	0	0	0
<b>B</b>	0	0	0	1	1	1
<b>C</b>	0	1	1	0	0	1
<b>D</b>	0	1	1	1	1	0
<b>E</b>	1	0	1	0	1	1
<b>F</b>	1	0	1	1	0	0
<b>G</b>	1	1	0	0	1	0
<b>H</b>	1	1	0	1	0	1

Esta combinação dos níveis dos atributos em cada alternativa baseou-se tanto nos arranjos ortogonais de Taguchi (maiores detalhes, ver em SOUZA, 1999) como na teoria de previsão de demanda de Kocur, Adler e Aunet (1982), já que ambos produzem delineamentos semelhantes. O delineamento dos demais conjuntos utilizados nesse experimento também tiveram como base essas duas teorias bastante utilizadas e reconhecidas no meio acadêmico.

As alternativas que resultaram do delineamento proposto foram então distribuídas em blocos, pois partiu-se do pressuposto que um experimento com 8 alternativas e 6 atributos pudesse gerar grandes erros de interpretação e fadiga da parte dos entrevistados. Optou-se então por dividir o conjunto de 8 alternativas em grupos de 4, utilizando-se da técnica de blocos incompletos balanceados (BIB). A tabela 6.3 apresenta a forma de divisão em 7 blocos com 4 alternativas cada (YATES, 1970).



Tabela 6.3 – Formação dos Blocos de Alternativas para o Conjunto 1

Blocos	Alternativas								
		A	B	C	D	E	F	G	H
	1	X	X	X	X				
	2	X	X			X	X		
	3	X	X					X	X
	4	X		X		X		X	
	5	X		X			X		X
	6	X			X	X			X
7	X			X		X	X		

Os cartões elaborados para a efetivação da pesquisa de campo foram construídos contendo ilustrações para auxiliar a compreensão dos entrevistados. Os cartões relativos a esse conjunto podem ser vistos no Anexo 5.

### 6.3.2. Delineamento Experimental do Conjunto 2

O Conjunto 2, intitulado Quanto às Restrições Consideradas, é composto pelos seguintes atributos:

- Número de bases: por base entende-se o ponto de saída e/ou chegada dos veículos, podendo esta ser única ou múltiplas.
- Tipo de veículo: alguns softwares só trabalham com roteirização considerando apenas um único tipo de veículo, enquanto que outros softwares consideram vários tipos de veículo.
- Duração máxima de um roteiro: em algumas situações essa pode ser uma variável relevante, principalmente no que diz respeito às leis trabalhistas ou à política de carga horária da empresa.
- Capacidade do veículo: neste caso considera-se como informação importante o limite máximo possível de embarque em termos de peso e/ou volume.
- Tamanho do veículo por cliente: alguns clientes têm restrições quanto ao tamanho do veículo que fará a entrega ou coleta de produtos (principalmente pelo fato de que certos veículos podem facilitar o processo de embarque/desembarque).

A especificação de cada nível dos atributos desse conjunto, os valores associados a esses e as variáveis representativas na função utilidade são mostradas na tabela 6.4.

Tabela 6.4 - Níveis dos atributos do Conjunto 2

<b>Atributo</b>	<b>Valoração</b>	<b>Níveis</b>	<b>Variável</b>
Número de bases (origem/destino)	0	Uma única	$X_{21}$
	1	Múltiplas	
Tipo de veículo	0	Único tipo	$X_{22}$
	1	Vários tipos	
Duração máxima de um roteiro	0	Não considera	$X_{23}$
	1	Considera	
Capacidade do veículo (Ton/m <sup>3</sup> )	0	Não considera	$X_{24}$
	1	Considera	
Tamanho do veículo por cliente	0	Não considera	$X_{25}$
	1	Considera	

Deste modo, a função utilidade local referente ao Conjunto 2 é dada por:

$$F(C2) = \alpha_{21}X_{21} + \alpha_{22}X_{22} + \alpha_{23}X_{23} + \alpha_{24}X_{24} + \alpha_{25}X_{25} \quad (6.2)$$

O delineamento experimental utilizado constitui-se em um ensaio fatorial fracionário  $2^3$  conforme consta na tabela 6.5. As alternativas ficam então definidas de acordo com as configurações dos níveis dos atributos.

Assim como feito para o conjunto 1, as alternativas que resultaram do delineamento proposto foram distribuídas em blocos. Optou-se então por dividir o conjunto de 8 alternativas em grupos de 4, utilizando-se novamente da técnica de blocos incompletos balanceados descrita por Yates (1970). A tabela 6.6 apresenta a forma de divisão em blocos para o conjunto 2.

Tabela 6.5 – Alternativas e níveis dos atributos do Conjunto 2

Alternativas	Atributos					
		$X_{21}$	$X_{22}$	$X_{23}$	$X_{24}$	$X_{25}$
	<b>A</b>	0	0	0	0	0
	<b>B</b>	0	0	0	1	1
	<b>C</b>	0	1	1	0	1
	<b>D</b>	0	1	1	1	0
	<b>E</b>	1	0	1	0	1
	<b>F</b>	1	0	1	1	0
	<b>G</b>	1	1	0	0	0
<b>H</b>	1	1	0	1	1	

Tabela 6.6 – Formação dos Blocos de Alternativas para o Conjunto 2

Blocos	Alternativas								
		<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>
	<b>1</b>	X	X	X	X				
	<b>2</b>	X	X			X	X		
	<b>3</b>	X	X					X	X
	<b>4</b>	X		X		X		X	
	<b>5</b>	X		X			X		X
	<b>6</b>	X			X	X			X
<b>7</b>	X			X		X	X		

Os cartões elaborados para a realização da pesquisa sobre esse conjunto podem ser vistos no Anexo 6.

### 6.3.3. Delineamento Experimental do Conjunto 3

O Conjunto 3, nomeado como Quanto à Interface e o Tratamento de Dados Reais, tem os seguintes atributos:

- Roteirização em tempo real: possibilidade de inserção de pontos de coleta ou entrega na rota que um veículo está fazendo, isso em tempo real.

- Mudança manual da solução: também conhecido como *drag and drop*, é a capacidade ou não de mudar a solução final dada pelo software, caso isso seja preciso (por exemplo, se for para privilegiar as exigências de algum cliente preferencial).
- Recursos gráficos: um software pode apresentar o roteiro apenas através de uma lista seqüencial, através de pontos ligados por linhas ou pode permitir a visualização da rota através de mapas e figuras explicativas.
- Cálculo real de distância e tempo: um software pode ou não ser capaz de considerar informações reais de tráfego, considerando, por exemplo, engarrafamentos e acidentes para a previsão do horário de chegada de uma mercadoria para um certo cliente.
- Interação com outros softwares: o software pode ou não ter alguma interação com outros programas computacionais, como rastreadores, WMS, GPS etc.

A especificação de cada nível desses atributos, bem como sua valoração e a variável associada são mostrados na tabela 6.7.

Tabela 6.7 - Níveis dos atributos do Conjunto 3

Atributo	Valoração	Níveis	Variável
Roteirização em tempo real	0	Impossível	$X_{31}$
	1	Possível	
Mudança manual da solução	0	Impossível	$X_{32}$
	1	Possível	
Recursos gráficos	0	Escassos	$X_{33}$
	1	Mapas/figuras	
Cálculo real de distância e tempo	0	Impossível	$X_{34}$
	1	Possível	
Interação com outros softwares	0	Nenhuma	$X_{35}$
	1	Existente	

Assim sendo, a função utilidade local referente ao Conjunto 3 é dada por:

$$F(C3) = \alpha_{31}X_{31} + \alpha_{32}X_{32} + \alpha_{33}X_{33} + \alpha_{34}X_{34} + \alpha_{35}X_{35} \quad (6.3)$$

Por ser composto por 5 atributos com 2 níveis cada, esse conjunto 3 tem o mesmo delineamento que o conjunto 2 (SOUZA, 1999). Assim, as alternativas e os respectivos níveis

e a formação de blocos de alternativas para esse conjunto seguem a mesma lógica descrita para o conjunto anterior. Isto pode ser verificado através das tabelas 6.8 e 6.9 respectivamente.

Tabela 6.8 – Alternativas e níveis dos atributos do Conjunto 3

	<b>Atributos</b>				
	$X_{31}$	$X_{32}$	$X_{33}$	$X_{34}$	$X_{35}$
<b>A</b>	0	0	0	0	0
<b>B</b>	0	0	0	1	1
<b>C</b>	0	1	1	0	1
<b>D</b>	0	1	1	1	0
<b>E</b>	1	0	1	0	1
<b>F</b>	1	0	1	1	0
<b>G</b>	1	1	0	0	0
<b>H</b>	1	1	0	1	1

Tabela 6.9 – Formação dos Blocos de Alternativas para o Conjunto 3

	<b>Alternativas</b>							
	<b>A</b>	<b>B</b>	<b>C</b>	<b>D</b>	<b>E</b>	<b>F</b>	<b>G</b>	<b>H</b>
<b>1</b>	X	X	X	X				
<b>2</b>	X	X			X	X		
<b>3</b>	X	X					X	X
<b>4</b>	X		X		X		X	
<b>5</b>	X		X			X		X
<b>6</b>	X			X	X			X
<b>7</b>	X			X		X	X	

Os cartões elaborados para a realização da pesquisa sobre esse conjunto podem ser vistos no Anexo 7.

### 6.3.4. Delineamento Experimental Geral

Seguindo a metodologia proposta, para cada uma das áreas de interesse (aqui, para cada um dos conjuntos), faz-se necessário aplicar um experimento de preferência declarada, havendo ainda a necessidade de aplicar mais um experimento que de alguma forma identifique a importância relativa de cada uma das áreas. Desse modo, foi importante também criar um quarto conjunto, o qual foi nomeado de “Conjunto 4 – Quanto ao software como um todo”, que tem essa função integradora. A tabela 6.10 apresenta os três atributos associados a esse conjunto, bem como seus níveis, respectiva valoração e variáveis associadas.

Tabela 6.10 - Níveis dos atributos do Conjunto 4

Atributo	Valoração	Representação	Variável
Capacidade algorítmica	0	Pequena	$F(C_1)$
	1	Grande	
Tratamento das restrições	0	Inserir poucas restrições	$F(C_2)$
	1	Inserir várias restrições	
Interface e tratamento de dados reais	0	Pouco satisfatório	$F(C_3)$
	1	Satisfatório	

Assim sendo, a função utilidade geral para avaliação de softwares para roteirização de veículos tem a seguinte expressão:

$$FU = \beta_1 F(C_1) + \beta_2 F(C_2) + \beta_3 F(C_3) \quad (6.4)$$

Neste caso, apesar de existirem somente 3 atributos, optou-se também por trabalhar com um delineamento mais simplificado. Ao invés de considerar o fatorial completo com as 8 alternativas, decidiu-se trabalhar com o fracionado, considerando assim apenas 4 alternativas (KOCUR, ADLER e AUNET; 1982). Essa decisão foi tomada pensando-se em não alongar ainda mais o experimento, já que o entrevistado, nessa altura dos trabalhos, já havia feito 21 escolhas (uma escolha para cada um dos 7 blocos de cada um dos 3 conjuntos). O delineamento experimental utilizado é apresentado na tabela 6.11.

Tabela 6.11 – Alternativas e níveis dos atributos do Conjunto 4

Alternativas	Atributos		
	$F(C_1)$	$F(C_2)$	$F(C_3)$
<b>A</b>	0	0	0
<b>B</b>	0	1	1
<b>C</b>	1	0	1
<b>D</b>	1	1	0

Os cartões utilizados na pesquisa de preferência declarada relativos a esse conjunto podem ser vistos no Anexo 8.

#### **6.4. Realização das Entrevistas**

As entrevistas foram realizadas nos meses de janeiro e fevereiro de 2004, sendo entrevistados tanto pesquisadores da área de transportes quanto empresas do ramo de logística e transporte, principalmente distribuidoras e operadores logísticos.

No total foram realizadas 25 entrevistas, sendo que 13 foram realizadas face a face e as outras 12 foram realizadas eletronicamente (via e-mail); nesse segundo caso, o material era enviado ao entrevistado, que, posteriormente, reenviava um documento preenchido com suas respostas. O material informativo e o documento a ser respondido são apresentados no Anexo 9.

Aos entrevistados inicialmente eram apresentados os motivos da entrevista. A seguir era solicitado ao entrevistado que preenchesse algumas perguntas de cunho social e profissional, para melhor identificação dos entrevistados no processo de segmentação da amostra, caso isso fosse relevante no futuro. Na sequência era iniciada a apresentação sequenciada dos 3 primeiros conjuntos. Para cada um destes os cartões eram apresentados seguindo a divisão em blocos delineada, onde cada bloco de cartões era identificado através de uma cor. Era solicitado então ao entrevistado que escolhesse dentro de cada um dos blocos a sua alternativa preferida. Cada escolha era devidamente anotada em formulário próprio, pelo entrevistador, no caso das entrevistas face a face, ou pelo próprio entrevistado, quando o mesmo estava realizando a entrevista eletronicamente. Por fim eram apresentados ao entrevistado os cartões relativos ao Conjunto 4, sendo solicitado nesse momento que o mesmo

ordenasse os cartões segundo suas preferências e anotasse suas escolhas em local apropriado, no mesmo formulário.

Os resultados obtidos a partir das entrevistas e codificados de forma apropriada foram posteriormente inseridos em um programa computacional desenvolvido para ajustes de modelos de preferência declarada, no caso, o programa LMPC (SOUZA, 1999).

### **6.5. Calibração dos Parâmetros das Funções Utilidade**

Como dito anteriormente, os dados descrevendo as preferências dos usuários foram analisados utilizando o programa LMPC. Através deste, os parâmetros são calibrados pela máxima verossimilhança, usando o método de ajuste de Newton-Raphson. As funções utilidade objetos de estudo foram aquelas descritas nas equações de (6.1) a (6.4) apresentadas anteriormente nesse capítulo.

As tabelas 6.12, 6.13 e 6.14 apresentam os ajustes obtidos para cada uma das 3 funções utilidade locais referentes aos conjuntos 1, 2 e 3 respectivamente; enquanto que na tabela 6.15 pode ser visto o ajuste encontrado para a função utilidade geral. Em cada uma dessas tabelas são apresentados também testes estatísticos de extrema importância para a verificação da confiabilidade dos resultados obtidos, sendo os de principal relevância: teste *t-student*, *likelihood* e *rho*.

Tabela 6.12 – Resultados da calibração dos parâmetros do Conjunto 1

Atributo	Coeffic.	Erro	Teste t	IC. (t=2,5%)
Janelas de tempo	0,7370	0,2036	3,6202	[0,330 ; 1,144]
Tempo carga/descarga	1,2096	0,2229	5,4268	[0,764 ; 1,655]
Considera velocidade	0,9922	0,2337	4,2456	[0,525 ; 1,460]
Horário início/fim	1,2360	0,2266	5,4544	[0,783 ; 1,689]
Roteiros por veículo	0,9112	0,2278	4,0004	[0,456 ; 1,367]
Restrições circulação	1,4139	0,2483	5,6942	[0,917 ; 1,911]
-----				
Eficiência =	0,8571			
Número de Entrevistas =	175		Número de Casos =	175
F(Betas_0) =	-242,6015		F(Betas_1) =	-163,9559
F(Blocos_0) =	-340,5343		F(Blocos_1) =	-346,1037
<b>LR (-2[F(0)-F(B)])=</b>	<b>157,2912</b>		LR (Blocos) =	-11,1389
<b>Rho =</b>	<b>0,3242</b>		Rho (Ajt) =	0,2994



Tabela 6.13 – Resultados da calibração dos parâmetros do Conjunto 2

Atributo	Coefic.	Erro	Teste t	IC. (t=2,5%)
Número de bases	0,9164	0,2751	3,3312	[0,366 ; 1,467]
Tipo de veículo	1,5842	0,2751	5,7588	[1,034 ; 2,134]
Duração máxima roteiro	1,6223	0,3054	5,3125	[1,012 ; 2,233]
Capacidade do veículo	1,4057	0,2343	5,9996	[0,937 ; 1,874]
Tamanho veículo/cliente	0,5018	0,2343	2,1417	[0,033 ; 0,970]
-----				
Eficiência = 0,8571				
Número de Entrevistas = 175		Número de Casos = 175		
F(Betas_0) = -242,6015		F(Betas_1) = -138,4352		
F(Blocos_0) = -340,5343		F(Blocos_1) = -351,5697		
<b>LR (-2[F(0)-F(B)])= 208,3326</b>		LR (Blocos) = -22,0709		
<b>Rho = 0,4294</b>		Rho (Ajt) = 0,4088		

Tabela 6.14 – Resultados da calibração dos parâmetros do Conjunto 3

Atributo	Coefic.	Erro	Teste t	IC. (t=2,5%)
Roteiriza tempo real	0,9344	0,2140	4,3672	[0,507 ; 1,362]
Mudança manual	1,0546	0,2140	4,9287	[0,627 ; 1,483]
Recursos gráficos	1,2689	0,2577	4,9240	[0,754 ; 1,784]
Cálculo real dist/tpo	1,2164	0,2303	5,2828	[0,756 ; 1,677]
Interação softwares	1,0562	0,2303	4,5869	[0,596 ; 1,517]
-----				
Eficiência = 0,8571				
Número de Entrevistas = 175		Número de Casos = 175		
F(Betas_0) = -242,6015		F(Betas_1) = -158,2787		
F(Blocos_0) = -340,5343		F(Blocos_1) = -347,7771		
<b>LR (-2[F(0)-F(B)])= 168,6456</b>		LR (Blocos) = -14,4857		
<b>Rho = 0,3476</b>		Rho (Ajt) = 0,3270		

Tabela 6.15 – Resultados da calibração dos parâmetros do Conjunto 4

Atributo	Coefic.	Erro	Teste t	IC. (t=2,5%)
Capacidade algorítmica	1,3313	0,4082	3,2616	[0,515 ; 2,148]
Tratamento das restrições	1,9316	0,4246	4,5496	[1,082 ; 2,781]
Interface e dados reais	1,3958	0,4059	3,4388	[0,584 ; 2,208]
-----				
Eficiência = 1,0000				
Número de Entrevistas = 25		Número de Casos = 75		
F(Betas_0) = -79,4513		F(Betas_1) = -45,2914		
<b>LR (-2[F(0)-F(B)])= 68,3199</b>		Rho (Ajt) = 0,3922		
<b>Rho = 0,4299</b>				

A aplicação do teste *t-student* revela um grau de confiabilidade de, no mínimo, 98% para todos os 19 atributos analisados, já que em todos os conjuntos o número de casos, neste caso, o número de entrevistas, é maior do que 30. Na expressão  $-2[F(0)-F(B)]$ ,  $F(0)$  representa o valor da função log – verossimilhança (*log – likelihood*) para todos os coeficientes nulos e  $F(B)$  representa o valor otimizado da função e é determinado usando a

distribuição  $\chi^2$ . Os valores obtidos com este teste, para cada um dos 4 conjuntos, indicam que, em todos os casos, é possível rejeitar a hipótese da nulidade de todos os parâmetros simultaneamente. Por fim, o teste estatístico do *rho* ( $\rho^2$ ) apresenta um ótimo desempenho em todos os 4 conjuntos de análise, tendo valores entre 0,3 e 0,43, valores apontados como excelentes, de acordo com Ortuzar e Willumsen (1994).

Os coeficientes ajustados, quando inseridos na expressão (6.4), geram a função utilidade geral, dada por:

$$FU = 1,3313F(C_1) + 1,9316F(C_2) + 1,3958F(C_3) \quad (6.5)$$

sendo que

$$F(C1) = 0,7370X_{11} + 1,2096X_{12} + 0,9922X_{13} + 1,2360X_{14} + 0,9112X_{15} + 1,4139X_{16} \quad (6.6)$$

$$F(C2) = 0,9164X_{21} + 1,5842X_{22} + 1,6223X_{23} + 1,4057X_{24} + 0,5018X_{25} \quad (6.7)$$

$$F(C3) = 0,9344X_{31} + 1,0546X_{32} + 1,2689X_{33} + 1,2164X_{34} + 1,0562X_{35} \quad (6.8)$$

Assim sendo, pode-se calcular então o valor da função utilidade geral máxima, partindo-se do pressuposto que todas as utilidades locais alcancem individualmente seus valores máximos (isto acontecendo quando todos os atributos respectivos atingirem seus níveis máximos de pontuação); o valor máximo da função utilidade geral equivalerá a 100% de conformidade com os atributos especificados. Sabendo que cada um dos atributos considerados pode assumir o valor máximo igual a 1, tem-se:  $FU_{\max} = 28,02113$  (considerar o valor inteiro 28 é uma aproximação bastante satisfatória).

Apresenta-se, a seguir, um exemplo ilustrativo de aplicação da metodologia.

## **6.6. Exemplo de Aplicação da Metodologia para Avaliação de um Software Roteirizador**

Para ilustrar a aplicação da metodologia proposta no presente trabalho, escolheu-se como objeto de avaliação um software de roteirização amplamente utilizado por empresas nacionais e internacionais (no Brasil o mesmo começou a ser comercializado no ano de 1993). Relatos de algumas empresas de distribuição que fazem uso do software apontam que a partir da adoção do mesmo obtiveram reduções significativas – de 10 a 25 por cento – no tempo do ciclo do pedido e no número de veículos de sua frota, conseguindo com isso que seus clientes diretos considerassem suas entregas mais eficientes.

O software, que aqui, por motivos éticos, vamos chamar de “xyz”, apresenta, à primeira vista, uma interface agradável e intuitiva, principalmente pelo fato de utilizar janelas que podem ser abertas e dimensionadas de acordo com as necessidades do usuário (nesse caso, o programador da empresa). Uma grande vantagem do software, segundo seu fabricante, é que o mesmo é facilmente integrado ao sistema corporativo adotado pela empresa.

O “xyz” possui uma série de recursos gráficos que incluem mapas detalhados da região de atuação, podendo inclusive apresentar na tela a posição do veículo durante a rota (bastando para tal uma conexão com sistema GPS). O software possibilita também que o programador crie, edite e atualize a malha viária a partir do *mouse*; ele permite ainda que o programador modifique rotas manualmente, caso isso seja necessário por alguma razão.

Segundo o fabricante do produto, com o uso do software a empresa pode esperar obter como resultados:

- A diminuição dos custos e da carga de trabalho;
- Uma melhora no gerenciamento da capacidade da frota;
- A minimização da distância percorrida, da duração das paradas e do uso de combustível; e
- Uma maior precisão na previsão (e cumprimento) dos horários de chegada e partida.

De fato, a partir da avaliação feita através da metodologia proposta, o software “xyz” apresenta resultados bastante satisfatórios, tanto em relação aos aspectos ergonômicos e de

qualidade (Módulo 1) quanto em relação aos critérios considerados importantes segundo usuários desse tipo de programa (Módulo 2).

Na avaliação dos critérios básicos, segundo o Módulo 1 da metodologia proposta, o avaliador responsável foi a própria autora do presente trabalho. O início da avaliação se deu a partir da associação dos pesos e posteriormente das respostas para cada pergunta de cada um dos 8 critérios relativos ao Módulo 1. Tendo sido feitos os devidos cálculos, o software foi considerado 80% conforme (requisito 2), sendo que os critérios com menor adequabilidade (requisito 1) foram Gestão de Erros, com 57,1% e Carga de Trabalho, com 60%. Já nos critérios 7 e 8, respectivamente, Eficiência e Funcionalidade, o software “xyz” foi considerado 100% adequado. Os cálculos mais detalhados, com a especificação dos pesos e das respostas para cada questão de cada critério estão detalhados no Apêndice B.

Tendo satisfeito os dois requisitos relativos ao Módulo 1, passou-se então para a avaliação dos critérios específicos dos softwares para roteirização de veículos. Esta segunda parte da avaliação (Módulo 2) se deu mais facilmente, apenas necessitando verificar, para cada um dos 16 atributos  $X_{ij}$  em que nível o software avaliado se insere, valendo aqui lembrar que o valor 1 é associado ao nível mais alto e o valor 0 é associado ao nível mais básico ou à inexistência do atributo no software.

A partir de pesquisas com um revendedor autorizado do software no Brasil, de questionamentos a algumas empresas usuárias e através do próprio conhecimento da autora, foi possível identificar a seguinte configuração para o software, de acordo com os níveis dos atributos relativos ao Módulo 2:

#### Conjunto 1 – Quanto à capacidade algorítmica do roteirizador

- Janelas de tempo: Flexíveis (valor 1)
- Tempo de carga e descarga: Não considera (valor 0)
- Considera a velocidade: Variável (valor 1)
- Horário de início e fim da viagem: Considera (valor 1)
- Roteiros por veículo: Múltiplos (valor 1)
- Restrições de circulação de veículos: Considera (valor 1)

Conjunto 2 - Quanto às restrições consideradas

- Número de bases (origem/destino): Múltiplas (valor 1)
- Tipo de veículo: Vários tipos (valor 1)
- Duração máxima de um roteiro: Considera (valor 1)
- Capacidade do veículo: Considera (valor 1)
- Tamanho do veículo por cliente: Não considera (valor 0)

Conjunto 3 – Quanto à interface e o tratamento de dados reais

- Roteirização em tempo real: Possível (valor 1)
- Mudança manual da solução: Possível (valor 1)
- Recursos gráficos: Mapas e figuras (valor 1)
- Cálculo real da distância e tempo: Possível (valor 1)
- Interação com outros softwares: Existentes (valor 1)

Desse modo, a partir das equações (6.6), (6.7) e (6.8), pôde-se calcular os valores das funções utilidades locais do software “xyz” :

$$F(C1) = 0,7370 \cdot 1 + 1,2096 \cdot 0 + 0,9922 \cdot 1 + 1,2360 \cdot 1 + 0,9112 \cdot 1 + 1,4139 \cdot 1 = 5,2903$$

$$F(C2) = 0,9164 \cdot 1 + 1,5842 \cdot 1 + 1,6223 \cdot 1 + 1,4057 \cdot 1 + 0,5018 \cdot 0 = 5,5286$$

$$F(C3) = 0,9344 \cdot 1 + 1,0546 \cdot 1 + 1,2689 \cdot 1 + 1,2164 \cdot 1 + 1,0562 \cdot 1 = 5,5305$$

A equação (6.5) expressa o valor da função utilidade global, já que associa a cada utilidade local um peso relativo. Sendo assim, para o caso específico do software avaliado, obteve-se o valor da função utilidade global igual a 25,44. Para finalizar a avaliação do software, bastou tomar como referência o valor máximo da função utilidade geral (28) e aplicar uma regra de três simples com o valor da função utilidade obtida para o software “xyz” (25,44).

De posse deste resultado, pode-se concluir que o software avaliado segundo a metodologia proposta está 90,86% em conformidade com os atributos específicos considerados essenciais para um software para roteirização de veículos.

## **CAPÍTULO 7 – CONCLUSÕES E RECOMENDAÇÕES**

Neste capítulo apresenta-se uma síntese dos resultados obtidos no decorrer do trabalho de pesquisa desenvolvido. Particularmente, salientam-se as contribuições que o presente trabalho trouxe para a área científica em que está inserido. Ao final do capítulo são sugeridos alguns pontos que poderão ser enfocados em pesquisas futuras.

### **7.1. Contribuições Gerais**

Com o presente trabalho de pesquisa propunha-se o desenvolvimento de uma metodologia para avaliação de software interativo que viesse a auxiliar a tomada de decisão quanto à aquisição de um software para a realização de uma tarefa específica.

Para alcançar tal objetivo fez-se necessária inicialmente uma pesquisa bibliográfica quanto às metodologias de avaliação e recomendações ergonômicas amplamente reconhecidas, divulgadas e utilizadas no meio acadêmico. Optou-se neste caso pelo aprofundamento em teorias oriundas de pesquisadores da corrente francesa de ergonomia, corrente esta que enfatiza mais significativamente questões referentes ao real funcionamento do sistema computacional na prática. As idéias de autores como Scapin e Bastien (1997), Barthet (1988) e Valentin, Vallery e Lucongsang (1993) foram devidamente estudadas e apresentam-se detalhadas no Capítulo 2 do presente texto. Com isso, uma primeira e importante contribuição do trabalho foi a sistematização do conhecimento relativo à ergonomia de software, enfatizando, como dito, as idéias de autores da corrente francófona de ergonomia.

Da mesma forma, para fundamentar a metodologia que se pretendia desenvolver, foram necessários levantamentos teóricos sobre as normas de qualidade existentes no Brasil e internacionalmente e que se referissem especificamente a questões quanto ao trabalho com sistemas informatizados. Tais normas – exclusivamente ISO e NBR – objetivam prescrever como um sistema para computadores deve se comportar, servindo também estas para sugerir novas práticas e novos *designs* para os programas. A segunda contribuição geral do trabalho então foi essa, qual seja, a reunião, em um único documento, das normas existentes quanto ao trabalho com sistemas informatizados. Tal contribuição é apresentada no Capítulo 3 do presente texto.

A terceira contribuição geral do trabalho desenvolvido e ora apresentado foi a verificação da possibilidade de se fazer uso das técnicas de preferência declarada como modo de inserir a opinião de especialistas e usuários em uma área onde não se encontra na bibliografia pesquisada aplicação alguma desse gênero. Imagina-se que este trabalho, por ser o pioneiro, pode incentivar outros pesquisadores a fazerem uso dessas técnicas para, de certa forma, “modelar” o usuário. As informações referentes a estas técnicas estão apresentadas no Capítulo 4 deste texto.

Por fim, pode-se dizer que a última contribuição geral do trabalho foi a elaboração e o desenvolvimento de uma metodologia inovadora para avaliação de softwares interativos, esta incluindo tanto aspectos ergonômicos e de qualidade de software quanto a opinião dos futuros usuários do mesmo.

## **7.2. Contribuições Específicas da Nova Metodologia**

Como descrito no Capítulo 5, a metodologia proposta é linear, composta basicamente por dois módulos. No primeiro módulo o software, objeto de avaliação, é analisado segundo suas características gerais sobre ergonomia e qualidade, enquanto que no segundo módulo são avaliadas suas peculiaridades, isto é, suas características específicas. Essa linearidade é importante pois se o software, por algum motivo, não for “aprovado” no Módulo 1, ou seja, se ele não satisfizer um dos dois requisitos explicitados, ele já pode ser considerado como não conforme, não havendo assim necessidade de uma avaliação mais detalhada, o que ocorre no Módulo 2. Assim, com tal linearidade adotada na metodologia, pode-se diminuir a carga de trabalho de um avaliador que, por exemplo, deseja avaliar vários softwares para um único propósito, visando escolher ao final apenas um como sendo, por assim dizer, o mais conforme.

O diferencial mais importante da metodologia proposta pode ser visto exatamente no Módulo 2. É neste módulo que se tem a possibilidade de contemplar a visão de especialistas e usuários da área a qual pertence o software interativo a ser avaliado. Isto se dá porque é no Módulo 2 que se faz uso das técnicas de preferência declarada, onde, a partir de entrevistas, consegue-se ter informações sobre as preferências dessas pessoas. Tais entrevistas realizadas, devidamente codificadas e processadas em programas computacionais apropriados, fornecem a informação não apenas de qual característica do software é mais importante, mas, mais

relevante, fornecem o peso relativo de cada característica (o dito atributo) considerada na pesquisa. Esta é uma das grandes vantagens da utilização da preferência declarada: poder trazer para dentro do planejamento as opiniões dos usuários.

Um outro diferencial da metodologia proposta é a possibilidade que ela dá de quantificar a avaliação. Apesar de ela tratar, tanto no Módulo 1 quanto no Módulo 2, de características mais subjetivas, optou-se por atribuir valores percentuais de conformidade em ambos o módulos da mesma. No Módulo 1 isto é feito a partir dos pesos (Muito Importante = 1,5, Importante = 1,0 e Não se aplica = 0) dados às questões e das respostas dadas às mesmas (Sim = 1, Parcialmente = 0,5 e Não = 0). Estes valores são utilizados para verificar o atendimento ou não dos requisitos do primeiro módulo. No Módulo 2 essa quantificação ocorre através de uma regra de três simples entre o valor máximo da função utilidade geral (que equivale a 100% de conformidade) e o valor atingido pelo software para essa mesma função (obtido, no caso, por uma função como a apresentada na expressão (5.2)).

Uma constatação que cabe no momento diz respeito à alta flexibilidade da metodologia proposta. Da forma como foi estruturada ela é perfeitamente adaptável a avaliação de qualquer tipo de software interativo. Apesar de o Módulo 1 ser fixo, isto é, ser composto de questões relativas aos 8 critérios ergonômicos e de qualidade previamente estipulados, essa adaptação dar-se-á através da ponderação dos pesos de cada uma dessas questões. Já o Módulo 2 é totalmente adaptável, haja vista que os atributos a serem considerados na pesquisa serão aqueles considerados de maior importância pela comunidade científica e/ou segundo levantamento preliminar com especialistas e usuários em potencial do software.

Resumidamente então, é possível concluir que as contribuições específicas da nova metodologia para avaliação de software interativo são:

- A linearidade da metodologia, com a verificação de requisitos mínimos a serem atendidos na primeira etapa de avaliação (Módulo 1);
- A inserção da visão (leia-se preferências) dos especialistas e usuários na avaliação do software (Módulo 2);
- A quantificação, em termos percentuais, da conformidade do software; e
- A alta flexibilidade da metodologia, podendo esta ser adaptada à avaliação de diferentes tipos de softwares interativos.



### **7.3. Contribuições Específicas da Aplicação da Metodologia Proposta**

Com a aplicação prática em roteirizadores de veículos, conforme apresentado no Capítulo 6, pretendeu-se, evidentemente, demonstrar a operacionalidade da metodologia proposta. Para viabilizar o desenvolvimento de tal aplicação, foi necessário de início um estudo amplo sobre a área na qual o software específico a ser avaliado se inseria, no caso, sobre transportes em geral e sobre roteirização de veículos em particular. Além disso, foi de extrema relevância a participação de especialistas em logística e de empresas de transporte que, de alguma forma, tem interesse na utilização de softwares para roteirização de veículos.

Considerando que as pessoas usam regras para simplificar suas decisões, agrupando características semelhantes em conjuntos e posteriormente utilizando regras para integrar tais conjuntos no momento da escolha final (Louviere e Timmermans, 1990), pode-se dizer que o delineamento experimental utilizado para o presente estudo permitiu contemplar o grande número de atributos considerados importantes na seleção de roteirizadores.

Os resultados obtidos na aplicação prática forneceram fortes evidências de que questões referentes às restrições consideradas pelo software são vistas com maior importância no momento da decisão quanto à adoção de um roteirizador de veículo. Mais especificamente, pôde-se perceber que os atributos “Tipo do Veículo”, “Duração Máxima do Roteiro” e “Capacidade do Veículo” são considerados os de maior peso na hora de tal decisão. Além disso, o conjunto de testes realizados comprova empiricamente a validade dos resultados alcançados.

Em nível geral, pode-se afirmar que os testes realizados evidenciam a aplicabilidade da metodologia, pois segundo os resultados obtidos a partir deles, não existiram discrepâncias significativas nas respostas às entrevistas realizadas, sugerindo então que o experimento foi bem delineado e que a metodologia é bem consistente.

Uma das maiores dificuldades na utilização de metodologias de avaliação de software é a definição (e quantificação) do que seja um software conforme. A aplicação da metodologia mostrou que esta última tem seu mérito nesse sentido. O exemplo realizado – a avaliação do software renomeado pela autora de “xyz” – mostra que, de fato, o referido software pode ser considerado bastante conforme (a um percentual de 90,86). De fato,

segundo informações obtidas do mercado consumidor, esse é um dos softwares para roteirizar veículos que mais tem ganhado espaço no mercado, devido exatamente à sua funcionalidade, sua alta eficiência e boa interface.

Por fim, pode-se ressaltar ainda como contribuição específica da aplicação da metodologia, a importância que os resultados obtidos têm para a área de transportes. Diversos autores, entre eles Cunha (2003) e Melo e Ferreira Filho (2001) enfatizaram em seus trabalhos a necessidade de identificar quais os critérios que devem ser adotados na seleção de softwares para roteirização de veículos, além da pouca discussão na literatura sobre tal questão. A metodologia proposta no presente trabalho, a partir de sua aplicação específica para esse tipo de software, trouxe essa contribuição significativa para a área, não apenas elencando os critérios a serem considerados, mas, mais ainda, dando valores significantes do grau de importância de cada um deles para o processo de escolha.

#### **7.4. Sugestões para Pesquisas Futuras**

De acordo com as conclusões apresentadas no item anterior, a aplicação desenvolvida trouxe importantes contribuições que devem propiciar um maior entendimento da metodologia proposta, comprovando assim sua robustez teórica e operacionalidade. Sugere-se, no entanto, a realização de novas aplicações, estas podendo ser feitas para outros tipos de softwares como também para outras áreas de estudo. Através dessas diferentes aplicações poderia ser possível fazer uma análise mais profunda no que diz respeito à performance obtida com a metodologia proposta no presente trabalho.

Apesar de existirem indícios de que o processo de escolha linear aditivo compensatório seja a base do comportamento de escolha do ser humano (FOERSTER, 1979, p.23), existem também outros modelos de decisão para seleção de alternativas, mais especificamente os modelos não compensatórios (FOERSTER, 1979; GOLOB e RICHARDSON, 1981; KAWAMOTO, 1994) que apresentam potencial e poderiam ser testados em uma metodologia deste tipo.

Outra sugestão seria a comparação dos resultados obtidos com a aplicação da referida metodologia com resultados oriundos da avaliação do mesmo software, porém sendo aplicada

outra metodologia de avaliação, esta, claro, devendo ser uma metodologia reconhecida no meio acadêmico e comprovadamente adequada para tal.

Uma proposta para melhorar os resultados encontrados com a aplicação específica da metodologia seria o aumento da amostra populacional de entrevistados nas pesquisas de preferência declarada. Apesar de os resultados obtidos terem sido bastante satisfatórios, o aumento da amostra pode ser interessante, principalmente se for desejável segmentá-la. Tal segmentação pode ser interessante para identificar, por exemplo, se há diferenças significativas nas respostas dadas por pessoas que trabalham em empresas ou naquelas fornecidas por especialistas na área, ou ainda, se há variáveis que são relevantes para aqueles que já fazem uso de roteirizadores mas que não são de extrema importância para os que ainda não usam, porém desejam fazê-lo no futuro. Com a amostra utilizada no presente trabalho, composta por 25 entrevistados, sendo que destes 14 são trabalhadores de empresas e 11 são pesquisadores especialistas na área, os resultados obtidos através de sua segmentação em empresas e pesquisadores não foram muito satisfatórios. Os testes estatísticos realizados não apresentaram bons resultados para todos os atributos (ver Anexo 10). Isso aconteceu, provavelmente, pelo fato da amostra, assim, não ser de tamanho suficiente e por isso então sugere-se a ampliação da amostra da população entrevistada caso se deseje fazer uma análise mais detalhada do problema aqui tratado.

Por fim, também sugere-se, para uma aplicação mais completa e verossímil da metodologia, levar em consideração com mais detalhamento as recomendações dadas pelos autores da área de preferência declarada quanto a questões de correlação entre alternativas, comprovando, através da matriz de variâncias-covariâncias, que elas são independente e identicamente distribuídas (mais detalhes ver, por exemplo, em LOUVIERE, HENSHER e SWAIT, 2000)

## REFERÊNCIAS

ALMEIDA, L. M. W. de. **Desenvolvimento de uma metodologia para análise locacional em sistemas educacionais usando modelos de interação espacial e indicadores de acessibilidade**. 1999. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 1999.

ALMEIDA, L. M. W. de; GONÇALVES, M. B. *A methodology to incorporate behavioral aspects in trip-distribution models with an application to estimate student flow*. **Environment and Planning A**, Pion Publication, Great Britain, v. 33, p. 1125-1138, 2001.

\_\_\_\_\_. Determinação de índices de acessibilidade a serviços escolares. In: CONGRESSO PAN AMERICANO DE ENGENHARIA DE TRÂNSITO E TRANSPORTE, 2000, Gramado. **Anais ...** Gramado: ANPET, 2000. p. 453-465.

\_\_\_\_\_. Uma avaliação do mérito de escolas de nível médio através da técnica de preferência declarada. **Revista Ensaio**, Rio de Janeiro, v. 7, n. 23, p. 201-218, 1999.

ASSAD, A. A. *Modeling and implementation issues in vehicle routing*. **Vehicle Routing: Methods and Studies**. North Holland, Amsterdam, p. 7-46, 1988.

ASSOCIAÇÃO BRASILEIRA DE ERGONOMIA. Disponível em: <http://www.abergo.org.br>. Acesso em: 14 janeiro 2003.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 8402**: Gestão da Qualidade e garantia da qualidade: terminologia. Rio de Janeiro, 1996.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 13596**: Tecnologia de informação: Avaliação de produto de software: Características de qualidade e diretrizes para o seu uso. Rio de Janeiro, 1996.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12119**: Tecnologia de informação: Pacotes de software: Testes e requisitos de qualidade. Rio de Janeiro: 1998.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-1**: Tecnologia de informação: Avaliação de produto de software: Parte 1: Visão geral. Rio de Janeiro, 2001.

ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 14598-5**: Tecnologia de informação: Avaliação de produto de software: Parte 5: Processo para avaliadores. Rio de Janeiro, 2001.

BALLOU, R. H. **Gerenciamento da Cadeia de Suprimentos** – Planejamento, Organização e Logística Empresarial. 4ª ed. Porto Alegre: Bookman, 2001.

BARRETO JÚNIOR, J. **Qualidade de Software**. Disponível em: <http://www.barreto.com.br/qualidade>. Acesso em: 02 agosto 2000.

BARTHET, M. F. *Logiciels Interactifs et Ergonomie*. Paris: Dunod, 1988.

BASTIEN, J. M. C.; SCAPIN, D. L.; LEULIER, C. *The ergonomic criteria and the ISO/DIS 9241-10 dialogue principles: a pilot comparison in an evaluation task*. *Interacting with Computers*, Elsevier, The Netherlands, v. 11, n. 3, p. 299-322, 1999.

BASTIEN, J. M. C.; SCAPIN, D. L. *Evaluating a User Interface with Ergonomic Criteria*. *International Journal of Human-Computer Interaction*, Ablex Publishing Corporation, New Jersey, v. 7, n. 2, p. 105-121, 1995.

\_\_\_\_\_. *A validation of ergonomic criteria for the evaluation of human-computer interfaces*. *International Journal of Human-Computer Interaction*, Ablex Publishing Corporation, New Jersey, v. 4, n. 2, p. 183-196, 1992.

BASTOS, L. C. **Planejamento da rede escolar: uma abordagem utilizando preferência declarada**. 1994. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 1994.

BEEVIS, D. *Ergonomics—Costs and Benefits Revisited*. *Applied Ergonomics*, Elsevier, n. 34, p. 491-496, 2003.

BEN-AKIVA, M.; LERMAN, S. *Discrete choice analysis: theory and application to travel demand*. The Mit Press: Massachusetts, 1985.

BEN-AKIVA, M.; MORIKAWA, T. *Estimation of switching models from revealed preferences and stated intentions*. *Transport Research*, Pergamon Press, Great Britain, v. 24A, n. 6, p. 485 – 495, 1990.

BHAT, C. R.; CASTELAR, S. *A unified mixed logit framework for modeling revealed and stated preferences: formulation and application to congestion pricing analysis in the San Francisco Bay área*. *Transportation Research – Part B*, Pergamon, London, v. 36, p. 593-616, 2002.

BODIN, L. D. *et al. Routing and scheduling of vehicles and crews: the state of art*. *Computers and Operations Research*, Pergamon Press, Great Britain, v. 10, n. 2, p. 63-212, 1983.

BODIN, L. D. *Twenty years of routing and scheduling*. *Operations Research*, v. 38, n. 4, p. 571-579, 1990.

BOWERSOX, D. J.; CLOSS, D. J. **Logística Empresarial**. São Paulo: Atlas, 2001.

BURKHARDT, J. *et al. L'apport de l'ergonomie aux NTIC éducatives: la formation médicale virtuelle*. In: NTICF'98 - Nouvelles Technologies de l'Information et de la Communication dans les Formations d'ingénieurs et dans l'industrie, 1998, Rouen. **Anais... Rouen: INSA de Rouen**, 1998, p.15-20.

CAMARGO, O. **Identificação dos principais atributos considerados no transporte de cargas: estudo de caso no oeste paranaense**. 2000. Dissertação. (Mestrado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 2000.

CAMPOS, F. C. A.; PASSOS, M. C. J. F.; VILLELA, P. R. C. **Software Agropecuário: A Busca de um Padrão de Qualidade.** Disponível em: <http://agrosoft.com/concurso/artigo.htm>. Acesso em: 23 setembro 2002.

CONSTANTINO, A. A. **Otimização de escala de trabalho para condutores de trem:** seqüenciamento de tarefas e alocação baseada em preferência declarada. 1997. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 1997.

COUTAZ, J. *Interfaces homme-ordinateur.* Paris: Dunod, 1990.

CUNHA, C. B. **Aspectos Práticos da Aplicação de Modelos de Roteirização de Veículos a Problemas Reais.** Disponível em: [www.ptr.poli.usp.br/ptr/docentes/cbcunha/files/roteirizacao\\_aspectos\\_praticos\\_CBC.pdf](http://www.ptr.poli.usp.br/ptr/docentes/cbcunha/files/roteirizacao_aspectos_praticos_CBC.pdf). Acesso em: 03 junho 2003.

CYBIS, W. *et al.* Uma abordagem ergonômica para o desenvolvimento de Sistemas Interativos. In: WORKSHOP SOBRE FATORES HUMANOS EM SISTEMAS COMPUTACIONAIS, 1998, Maringá. **Anais...** Maringá: 1998.

CYBIS, W. **A identificação dos objetos de interfaces homem - computador e seus atributos ergonômicos.** 1994. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 1994.

DALE, C.J.; VAN DER ZEE, H. *Software productivity metrics: who needs them?* **Information and Software Technology.** France, v. 34, n. 11, p. 731-738, 1992.

EKLUND, J. *Ergonomics, quality and continuous improvement: conceptual and empirical relationship in an industrial context.* **Ergonomics,** Taylor & Francis, London, v. 40, n. 10, p. 982 – 1001, 1997.

*ERGONOMICS.* London: Taylor and Francis, v. 41, n.11, 1998.

FOERSTER, J. F. *Mode choice decision process models: a comparison of compensatory and non-compensatory structures.* **Transportation Research,** London, n. 13A, p. 17-28, 1979.

FOWKES, T.; WARDMAN, M. *Design of SP Travel Choice Experiments, with special reference to Taste Variation.* **Journal of Transport Economics and Policy.** The London School of Economics and Political Science and the University of Bath, London, v. XXII, n. 1, p. 27-44, 1988.

GAINES, B. R.; SHAW, M. L. G. **A Interação Computador - Usuário: Um novo meio de comunicação.** Rio de Janeiro: LTC Editoras S. A., 1987.

GAMEZ, L. **Ergonomia Escolar e as Novas Tecnologias no Ensino: Enfoque na Avaliação de Software Educacional.** 1998. Dissertação. (Mestrado em Engenharia Humana) - Universidade do Minho. Guimarães, 1998.

GOLDEN, B.; BODIN, L.; GOODWIN, T. *Microcomputer-based vehicle routing and scheduling software*. *Computers and Operations Research*, v. 13, n. 2/3, p. 277-285, 1986.

GOLOB, T. G.; RICHARDSON, A.J. *Noncompensatory and discontinuous constructs in travel-behavior models*. In: *New horizons in travel behavior research*, Lexington, Massachusetts, 1981.

GOMES, L. F. M.; OLIVEIRA, J. R. de. **Análise de estratégias para aumento de qualidade e produtividade em informática**: aplicação de auxílio multicritério à decisão. Rio de Janeiro: Qualitymark, 1993.

GOVERNEMENT DU QUÉBEC. *Guide de conception d'interactions personnes-systèmes (IPS) – Méthode de conception ergonomique d'IPS (Les orientations technologiques du réseau sociosanitaire)*. Ministère de la Santé et des Services Sociaux, Québec, 1997.

HALL, R. *Vehicle Routing Software Survey*. Disponível em: [www.lionhrtpub.com/orms/surveys/Vehicle\\_Routing/vrss.html](http://www.lionhrtpub.com/orms/surveys/Vehicle_Routing/vrss.html). Acesso em: 07 dezembro 2002.

HALL, R.; PARTYKA, J. *On the Road to Efficiency*, 1997. Disponível em: <http://www.lionhrtpub.com/orms/orms-6-97/toc.html>. Acesso em: 07 dezembro 2002.

HENDRICK, H. *Determining the cost-benefits of ergonomics projects and factors that lead to their success*. *Applied Ergonomics*. Elsevier, n. 34, p. 419-427, 2003.

HIX, D.; SCHULMAN, R. S. *Human-Computer interface development tools: a methodology for their evaluation*. *Communications of the ACM*. London, v. 34, n. 3, p. 74-87, 1991.

HUMAN FACTORS AND ERGONOMICS SOCIETY. Disponível em: <http://hfes.org>. Acesso em: 20 janeiro 2003.

IIDA, I. *Ergonomia: Projeto e Produção*. 4. ed. São Paulo: Editora Edgar Blücher Ltda, 1997.

ISO. **ISO 9241-1**: *Ergonomic requirements for Office work with visual display terminals (VDTs): Part 1: General Introduction*. Genève, 1992.

ISO. **ISO 9241-2**: *Ergonomic requirements for Office work with visual display terminals (VDTs): Part 2: Visual display requirements*. Genève, 1992.

ISO. **ISO 9241-10**: *Ergonomic requirements for Office work with visual display terminals (VDTs). Part 10: Dialogue principles*. Genève, 1996.

JONES, P. *An overview of stated preference techniques*. PTRC Course. p. 1-14, 1991.

KAWAMOTO, E. Calibração do modelo semi-compensatório de escolha modal. *Transportes*, Revista da ANPET, Rio de Janeiro, v. 2, n. 1, 31-41, 1994.

KOCUR, G.; ADLER, W.; AUNET, B. *Guide to forecasting travel demand with direct utility assessment*. Technical Report. U. S. Department of Transportation. UMTA-NH-11-0001-82-1, 1982.

KROES, E. P. e SHELDON, R. J. *Stated preference methods*. **Journal of Transport Economics and Policy**, The London School of Economics and Political Science and the University of Bath, London, v. XXII, n. 1, pp 11-20, 1988.

LAPORTE, G. *The Vehicle Routing Problem: An Overview of exact and approximate algorithms*. **European Journal of Operational Research**, North-Holland, v. 59, p. 345-358, 1992.

LOUVIERE, J. J. *Hierarchical Information Integration: A new method for the design and analysis of complex multiattribute judgment problems*. In: KINNEAR, T.C. (Ed.). **Advances in Consumer Research**, vol. XI. Association for Consumer Research, 1984. p. 148-155.

\_\_\_\_\_. *Conjoint analysis modeling of stated preference: a review of theory, methods, recent developments and the external validity*. **Journal of Transportation Economics and Policy**, The London School of Economics and Political Science and the University of Bath, London, v. XXII, n. 1, pp 93-119, 1988.

LOUVIERE, J.; TIMMERMANS, H. *A review of recent advances in decompositional preference and choice models*. *Tijdschrift voor Econ. En Soc. Geografie*, v. 81, n. 3, p. 214-224, 1990.

LOUVIERE, J. J. , HENSHER, D. e SWAIT, J. **Stated Choice Methods**. Cambridge, Cambridge University Press, 2000.

MARQUES, K. W. B. **Preferência Declarada Aplicada à Alocação Ótima de Alunos às Escolas: Um estudo de caso**. 2003. Dissertação. (Mestrado em Ciências) - Universidade Federal do Paraná. Curitiba, 2003.

MEDEIROS, M. A. **ISO 9241: Uma Proposta de Utilização da Norma para Avaliação do Grau da Satisfação de Usuários de Software**. 1999. Dissertação. (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina. Florianópolis, 1999.

MELO, A. C. S.; FERREIRA FILHO, J. V. M. *Sistemas de Roteirização e Programação de Veículos*. **Pesquisa Operacional**. SOBRAPO, Rio de Janeiro, v. 21, n. 2, p. 223 – 232, 2001.

MOÇO, S. de S. **O uso de cenários como uma técnica de apoio para avaliações ergonômicas de softwares interativos**. 1996. Dissertação. (Mestrado em Engenharia de Produção) – Universidade Federal de Santa Catarina. Florianópolis, 1996.

NIELSEN, J. **Usability Engineering**. New Jersey: A P Professional, 1993.

NORTON, D. *Rainbow Usability: Customization vs Training in Developing Countries*. **Interactions**, ACM Press, v. 10, n. 2, p. 9-12, 2003.

NOVAES, A. G.; CARVALHO, M. C. *Market share modelling in intercity bus travel in Brazil*. In: 7<sup>th</sup> INTERNATIONAL CONFERENCE ON TRAVEL BEHAVIOR, 1994, Chile. **Anais...** Chile, 1994. v. 2, p. 577-588.



NOVAES, A. G. *et al.* Aferição do nível logístico - portuário por meio de técnicas de preferência declarada. In: CONGRESSO DE PESQUISA E ENSINO EM TRANSPORTES, 1996, São Paulo. **Anais...** São Paulo: ANPET, 1996. v. 2, pp 605-617.

NOVAES, A. G. **Logística e Gerenciamento da Cadeia de Distribuição**. Rio de Janeiro: Campus, 2001.

ORTUZAR, J. de D.; WILLUMSEN, L. G. *Modelling transportation*. 2. ed. John Wiley and Sons: Chichester England, 1994.

ORTUZAR, J. de D. *Modelos econométricos de elección discreta*. Ed. Universidad Católica de Chile: Santiago, 1999.

PALADINI, E. P. **Qualidade Total na Prática**: Implantação e avaliação de sistemas de qualidade total. 2. ed. Editora Atlas: São Paulo, 1997.

PARASURAMAN, R. *Designing automation for human use: empirical studies and quantitative models*. *Ergonomics*, Taylor & Francis, London, v. 43, n. 7, p. 931 – 951, 2000.

PARTYKA, J. G.; HALL, R. *On the Road to Service*, 2000. Disponível em: [www.lionhrtpub.com/orms/orms-8-00/vehiclerouting.html](http://www.lionhrtpub.com/orms/orms-8-00/vehiclerouting.html). Acesso em: 03 dezembro 2002.

PELIZARO, C.; DA SILVA, A. N. R. Uma avaliação comparativa de performance de algoritmos de software comerciais para roteirização de veículos. **Transporte em Transformação VII**, Ed. LGE, p. 181-198, 2003.

PRESSMAN, R. S. **Engenharia de Software**. São Paulo: Makron Books, 1995.

RAMOS, E. **Análise Ergonômica do Sistema hiperNet Buscando o Aprendizado da Cooperação e Autonomia**. 1995. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 1995.

RASKIN, J. *Summary of The Human Interface*. Disponível em: [http://humane.sourceforge.net/humane\\_interface/summary\\_of\\_thi.html](http://humane.sourceforge.net/humane_interface/summary_of_thi.html). Acesso em: 24 fevereiro 2004.

RAY, E. *User Interface Design*. New Jersey: Prentice-Hall, 1994.

REVISTA TECNOLÓGICA. São Paulo: Publicare, n. 91, junho 2003.

REZENDE, D. A. **Engenharia de Software e Sistemas de Informação**. Rio de Janeiro: Brasport, 1999.

RONEN, D. *Perspectives on practical aspects of truck routing and scheduling*. *European Journal of Operational Research*, n. 35, p. 137-145, 1988.

SCALET, D. Qualidade dos Produtos de Software. In: WEBER, K. C.; ROCHA, A. R. C. da; NASCIMENTO, C. J. do (Org.). **Qualidade e Produtividade em Software**. 4. ed. ren. São Paulo: Makron Books, 2001. p. 41-52.

SCAPIN, D.; BASTIEN, J. M. C. *Ergonomic criteria for evaluating the ergonomic quality of interactive systems. Behaviour & Information Technology*, Taylor & Francis, London, v. 16, n. 4/5, p. 220 - 231, 1997.

SENN, L. A. D. S. *Travellers willingness to pay for education in travel time variability. In: VI WORLD CONFERENCE ON TRANSPORTATION RESEARCH*, 1992, Lion. *Anais... Lion*, 1992.

SCHIMITZ, R. **Uma contribuição metodológica para avaliação da tarifa de pedágio em rodovias**. 2001. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 2001.

SHNEIDERMAN, B. *Designing the user interface*. 3. ed. USA: Addison-Wesley, 1998.

SOMMERVILLE, I. *Software Engineering*. 5. e. USA: Addison Wesley, 1997.

SOUZA, O. A. de. **Delineamento experimental em ensaios fatoriais utilizados em preferência declarada**. 1999. Tese. (Doutorado em Engenharia de Produção) - Universidade Federal de Santa Catarina. Florianópolis, 1999.

SPERANDIO, J. C. *L'ergonomie du travail mental*. Paris: Masson, 1983.

STANTON, N.; BABER, C. *On the cost-effectiveness of ergonomics. Applied Ergonomics, Elsevier*, n. 34, p. 407-411, 2003.

TEODORO, V. D. **Educação e Computadores**. GEP: Lisboa, 1992.

THE ERGONOMICS SOCIETY. Disponível em: <http://www.ergonomics.org.uk>. Acesso em: 14 janeiro 2003.

VALENTIN, A.; VALLERY, G.; LUCONGSANG, R. *L'evaluation ergonomique des logiciels: une démarche itérative de conception*. Montrouge: Anact, 1993.

VICENTE, K. *Cognitive Work Analysis – Toward Safe, Productive and Healthy Computer-Based Work*. United States of America: Lawrence Erlbaum Associates, 1999.

WEBER, K. C. Qualidade e Produtividade em Software. In: WEBER, K. C.; ROCHA, A. R. C. da (Org.). **Qualidade e Produtividade em Software**. 3. ed. São Paulo: Makron Books, 1999. p. 1 – 22.

YATES, F. *Incomplete randomized blocks. In: Experimental design: selected papers of Frank Yates*. London: Charles Griffin, 1970.

## **APÊNDICES**

## APÊNDICE A

### Distribuição de Gumbel

Anteriormente nesse texto foi salientado que a função utilidade randômica pode ser expressa como a soma de uma componente determinística e de uma componente aleatória, tendo então a expressão:

$$U_{in} = V_{in} + \varepsilon_{in}$$

onde  $U_{in}$  é a utilidade da alternativa  $i$  para o indivíduo  $n$ ,  $V_{in}$  é a parcela determinística da alternativa  $i$  para o indivíduo  $n$  e  $\varepsilon_{in}$  é a parcela aleatória associada à alternativa  $i$ . Nessa explanação assumiu-se que os termos  $\varepsilon_{i n}$  são independentes e identicamente distribuídos segundo uma função distribuição de probabilidade Gumbel.

A função:

$$F(\varepsilon) = e^{-e^{-\mu(\varepsilon-\eta)}}$$

representa a função distribuição acumulada de Gumbel, onde  $\eta$  é o parâmetro de localização e  $\mu$  é o parâmetro de escala ( $\mu > 0$ ). Essa distribuição tem como principais características:

- Sua média pode ser calculada por  $\frac{1}{\mu} \ln \sum_i e^{\mu\eta_i}$
- Sua variância pode ser calculada por  $\frac{\pi^2}{6\mu^2}$

Ben-Akiva e Lerman (1985, p. 105) salientam duas propriedades que proporcionam meios de se derivar o modelo logit binário e conseqüentemente, por sua generalização, o modelo logit multinomial: a primeira propriedade diz que a distribuição de Gumbel é preservada por transformações lineares; já a segunda propriedade mostra que a distribuição

Gumbel da diferença entre duas variáveis independentes tem o mesmo parâmetro de escala ( $\mu$ ).

Além dessas duas importantes propriedades, pode-se destacar que, se  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_i$  são variáveis aleatórias independentes e Gumbel distribuídas com parâmetros  $(\eta_1, \mu), (\eta_2, \mu), \dots, (\eta_i, \mu)$  respectivamente, então a variável dada como  $\max(\varepsilon_1, \varepsilon_2, \dots, \varepsilon_i)$  é também Gumbel distribuída, tendo como parâmetros

$$\left( \frac{1}{\mu} \ln \sum_i e^{\mu \eta_i}, \mu \right).$$

Essa consideração também tem papel bastante significativo no desenvolvimento do modelo logit multinomial. Esse desenvolvimento pode ser verificado com maiores detalhes nas obras de Ben-Akiva e Lerman (1985) e Ortúzar (2000).

## APÊNDICE B

### Avaliação do Software de Roteirização “xyz” a partir da metodologia proposta

#### A) Associação dos Pesos e Respostas Para Cada Pergunta de Cada Critério do Módulo 1

##### Pesos:

$p_i = 1,5$  – questão muito importante

$p_i = 1,0$  – questão importante

$p_i = 0$  – questão não se aplica

##### Respostas:

S = 1,0 – sim

P = 0,5 – praticamente

N = 0 – não

#### **Critério 1. Dispositivos de Apresentação e Linguagem de Interação**

- Q1.1) O vocabulário usado é familiar ao usuário e está de acordo com a tarefa a ser realizada?

**Peso ( 1,5 ) Resposta ( 0,5 )**

- Q1.2) O vocabulário e as abreviações usadas são facilmente compreensíveis e, se necessário, de fácil memorização?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q1.3) Um comando, botão ou opção de menu, quando selecionado, realiza sempre a mesma ação?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q1.4) As informações na tela são legíveis, padronizadas e bem distribuídas espacialmente?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q1.5) O tamanho da fonte e cores são usados adequadamente?

**Peso ( 1 ) Resposta ( 1 )**

- Q1.6) As informações da mesma natureza (entrada de dados, botões, textos e títulos) aparecem sempre no mesmo local e seguem o mesmo padrão em todas as telas?

**Peso ( 1 ) Resposta ( 1 )**

- Q1.7) Todas as entradas de dados feitas pelo usuário são mostradas na tela, mesmo que através de símbolos como, por exemplo “\*”?  
**Peso ( 1 ) Resposta ( 1 )**

- Q1.8) O software fornece informações sobre o estado do processamento das informações quando esse é muito longo?  
**Peso ( 1 ) Resposta ( 1 )**

- Q1.9) É possível distinguir visualmente, através de pastas, por exemplo, áreas que tenham diferentes funções?  
**Peso ( 1 ) Resposta ( 1 )**

## **Critério 2. Controle do Usuário**

- Q2.1) O usuário precisa sempre confirmar, através da tecla ENTER, a inicialização do processamento das informações e a efetivação das entradas de comandos?  
**Peso ( 1,5 ) Resposta ( 1 )**

- Q2.2) O usuário pode cancelar uma ação sempre que desejar, voltando assim à situação anterior?  
**Peso ( 1,5 ) Resposta ( 1 )**

- Q2.3) O usuário pode interromper uma tarefa e retomá-la posteriormente do mesmo ponto?  
**Peso ( 1 ) Resposta ( 1 )**

- Q2.4) O usuário pode controlar a velocidade de interação, principalmente no que diz respeito à entrada de dados?  
**Peso ( 1 ) Resposta ( 0,5 )**

- Q2.5) O usuário pode controlar o que aparece na tela, desde o posicionamento do cursor até a quantidade de informações apresentadas?  
**Peso ( 1,5 ) Resposta ( 1 )**

- Q2.6) O usuário pode ter acesso a todas as informações a qualquer momento?  
**Peso ( 1,5 ) Resposta ( 1 )**

## **Critério 3. Adaptabilidade**

- Q3.1) O software permite que o usuário configure a tela e os dispositivos de entrada e saída de dados de acordo com suas preferências e necessidades?  
**Peso ( 1,5 ) Resposta ( 1 )**

- Q3.2) O software disponibiliza diferentes tipos de diálogo para diferentes níveis de habilidade e experiência do usuário (principalmente diferenciando iniciantes e experientes)?

**Peso ( 0 ) Resposta ( 0,5 )**

- Q3.3) O software prevê navegação passo-a-passo para iniciantes e teclas de atalho para os mais experientes?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q3.4) O software dispõe informações diferenciadas, inclusive sobre erros ocorridos, de acordo com o nível de conhecimento do indivíduo?

**Peso ( 0 ) Resposta ( 0,5 )**

- Q3.5) O software permite que o usuário inclua comandos personalizados e configure parâmetros de tempo operacional de acordo com suas necessidades individuais?

**Peso ( 1 ) Resposta ( 0,5 )**

#### **Critério 4. Gestão de Erros**

- Q4.1) O software solicita a confirmação de uma ação sempre que essa tenha sérias consequências, como, por exemplo, a perda de dados ou o não processamento de alguma informação?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q4.2) O software protege funções e comandos perigosos?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q4.3) O software apresenta falhas frequentemente?

**Peso ( 1,5 ) Resposta ( 0,5 )**

- Q4.4) O software é capaz de recuperar dados no caso de ocorrência de falhas?

**Peso ( 1,5 ) Resposta ( 0,5 )**

- Q4.5) O software apresenta uma mensagem explicativa, indicando a opção correta, quando o usuário seleciona uma função inválida?

**Peso ( 1 ) Resposta ( 0 )**

- Q4.6) As mensagens de erro são claras, instruindo o usuário para a correção do problema?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q4.7) O software, quando solicitado, fornece explicações adicionais durante a correção de erros?

**Peso ( 1 ) Resposta ( 0 )**



- Q4.8) O software permite que o usuário corrija os erros direta e imediatamente, sem ter que mudar o estado atual do sistema?

**Peso ( 1 ) Resposta ( 1 )**

- Q4.9) O software permite que o usuário corrija somente os comandos ou a parte digitada incorretamente?

**Peso ( 1 ) Resposta ( 1 )**

### **Critério 5. Carga de Trabalho**

- Q5.1) Dados que podem ser derivados de outros já acessíveis ao sistema são obtidos automaticamente?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q5.2) O software pode importar dados de uma tela para outra, evitando assim a memorização do usuário?

**Peso ( 1,5 ) Resposta ( 0,5 )**

- Q5.3) O software disponibiliza somente as informações necessárias para aquele momento?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q5.4) O software possibilita o acesso direto a um ponto específico do mesmo, evitando assim toda uma navegação até esse?

**Peso ( 1,5 ) Resposta ( 0,5 )**

- Q5.5) O software exibe valores *default* apropriados?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q5.6) O software, quando possível, minimiza os passos na execução de uma tarefa?

**Peso ( 1 ) Resposta ( 0,5 )**

### **Critério 6. Usabilidade**

- Q6.1) É fácil aprender a usar o software?

**Peso ( 1 ) Resposta ( 1 )**

- Q6.2) Existem ferramentas como, por exemplo, tutoriais interativos, que auxiliam o aprendizado do software?

**Peso ( 1 ) Resposta ( 0 )**

- Q6.3) É fácil operar o software?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q6.4) O software dispõe de orientação e ajuda para a operação do software sempre que necessário?

**Peso ( 1 ) Resposta ( 0,5 )**

- Q6.5) O software dispõe de um glossário para auxiliar o usuário na compreensão de termos técnicos?

**Peso ( 1 ) Resposta ( 1 )**

### **Critério 7. Eficiência**

- Q7.1) O tempo de resposta do software é aceitável, considerando o “tamanho” da tarefa?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q7.2) Considerando a quantidade de recursos utilizados pelo software (impressora, outros produtos de software, manutenção e suporte), pode-se dizer que o mesmo é eficiente?

**Peso ( 1 ) Resposta ( 1 )**

### **Critério 8. Funcionalidade**

- Q8.1) O software se propõe a realizar a tarefa requerida?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q8.2) O software faz o que propõe corretamente, disponibilizando as funções adequadas para a realização da tarefa?

**Peso ( 1,5 ) Resposta ( 1 )**

- Q8.3) O software posiciona as funções nos menus respeitando o modo de trabalho dos usuários, isto é, colocando as opções mais utilizadas estrategicamente?

**Peso ( 1 ) Resposta ( 1 )**

- Q8.4) O software interage com outros sistemas essenciais na realização da tarefa?

**Peso ( 1 ) Resposta ( 1 )**

- Q8.5) O software está de acordo com normas, leis e padrões da área na qual se insere a aplicação?

**Peso ( 1 ) Resposta ( 1 )**

B) Verificação dos requisitos do Módulo 1

**Requisito 1:**  $VC_j \geq 30 \quad \forall j = 1, 2, \dots, 8$  com  $VC_j = \frac{\sum_{i=1}^k p_i a_i}{\sum_{i=1}^k p_i} \cdot 100$

De fato, o critério é válido para todos os 8 critérios, já que:

- $VC_1 = \frac{9,25}{10,5} \cdot 100 = 88,1\%$
- $VC_2 = \frac{7,5}{8} \cdot 100 = 93,7\%$
- $VC_3 = \frac{2,5}{3,5} \cdot 100 = 71,4\%$
- $VC_4 = \frac{6}{10,5} \cdot 100 = 57,1\%$
- $VC_5 = \frac{4,5}{7,5} \cdot 100 = 60\%$
- $VC_6 = \frac{4}{5,5} \cdot 100 = 72,7\%$
- $VC_7 = \frac{2,5}{2,5} \cdot 100 = 100\%$
- $VC_8 = \frac{6}{6} \cdot 100 = 100\%$

**Requisito 2:**  $\overline{VC_j} \geq 55$ , onde  $\overline{VC_j} = \frac{\sum_{j=1}^n VC_j}{n}$  é a média de todos os critérios  $VC_j$ .

Este requisito também é satisfeito pois:

$$\overline{VC_j} = \frac{88,1 + 93,7 + 71,4 + 57,1 + 60 + 72,7 + 100 + 100}{8} = \frac{643}{8} = 80,37\% .$$

Deste modo, como os dois requisitos foram satisfeitos simultaneamente, deve-se dar continuidade à avaliação do software, agora através do Módulo 2, onde se avaliam os critérios específicos, segundo levantamento obtido a partir de experimentos de preferência declarada.

**ANEXOS**

## ANEXO 1

### **Tipos de questões do pré-diagnóstico do ergonômista**

(VALENTIN, VALLERY e LUCONGSANG, 1993, p. 129-131 – tradução livre)

#### **1. Apresentação das informações**

- Existe uma estruturação espacial padrão das informações?
- Existe uma organização homogênea das informações para cada classe de aplicação ou cada família de menus?
- As informações importantes são claras e informativas?
- Os textos escritos são claros e informativos?
- O vocabulário e as abreviações são facilmente compreensíveis (verificar depois com o setor profissional)?
- Existe um isomorfismo na organização das informações nos diferentes suportes (papel/tela)?
- Os modos de diferenciação da informação na tela são coerentes?
- A utilização de “*inverse video*” e outros atributos são racionais?
- O número de cores é limitado?
- As cores facilmente indicam o agrupamento de informações?
- Uma cor tem sempre o mesmo significado?
- O número de janelas é limitado?
- O posicionamento de uma janela é sempre fácil?
- Algumas janelas se encontram recobertas? Isso é um problema para o tratamento das informações?
- A posição e o tamanho das janelas são variáveis? A partir de que critérios?

#### **2. Modos de captura (Modo de entrada de dados)**

- O software permite ignorar o funcionamento do material ou de alguns elementos técnicos (sistema de exploração, sistema de comandos, manipulação de disquetes, gestão do disco rígido...)?
- O modo de captura das informações é adaptado às necessidades (teclado, *mouse*, caneta luminosa...)?

- Existe uma lógica (ou uma coerência) nos menus para ter acesso aos tratamentos?
- Os símbolos e códigos de captura são explícitos e fáceis de memorizar (verificar depois com o setor profissional)?
- Existe uma correspondência entre a arborescência do menu e os modos de trabalho dos operadores (verificar com os usuários)?
- A estrutura dos menus sugere tal arborescência?
- O usuário é guiado pelas diferentes arborescências?
- O modo de captura convém tanto aos iniciantes quanto aos experientes?
- A estabilidade do modo de captura e das funções-toque são respeitadas (mesma sintaxe, mesmos efeitos)?

### **3. Funções perigosas**

- As funções ou comandos perigosos são protegidas?
- Existe um meio de anular a última operação?

### **4. Gestão de erros**

- As mensagens de erros são positivas e informativas; permitem resolver o incidente ou identificar os modos de ajuda?
- A retomada dos erros é possível, satisfatória?

### **5. Documentação**

- A documentação em papel é clara, suficiente, operatória?
- A documentação na tela é clara, suficiente, operatória?

### **6. Flexibilidade**

- O produto pode ser personalizado?
- O modo de diálogo é diferente para os iniciantes e para as pessoas experientes?

## ANEXO 2

### **Entrevista sobre o software, com os usuários**

(VALENTIN, VALLERY e LUCONGSANG, 1993, p. 135-142 – tradução livre)

O presente questionário tem por objetivo verificar a adequação do software às tarefas demandadas, antes e após a formação.

Para cada uma das duas situações de uma apreciação codificada de 1 (muito ruim) à 5 (muito bom), circulando o número que parecer conveniente segundo sua experiência.

Sempre que você puder, justifique sua resposta.

Observação: as pessoas que não receberam a formação só devem responder a primeira parte de cada questão.

#### **1. Qual sua apreciação geral sobre a facilidade de utilização do software?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

#### **2. O software permite realizar corretamente as tarefas previstas por seu intermediário?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

#### **3. O software permite ignorar o funcionamento do material (sistema de exploração, sistema de comandos, manipulação de disquetes, gestão do disco rígido...)?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**4. O que você pensa sobre as informações que lhe são apresentadas na tela?****• O vocabulário usado corresponde ao de sua profissão?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**• Os símbolos e códigos apresentados são explícitos, compreensíveis?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**• As informações e abreviações são claras (não ambíguas)?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**• É possível situar ou encontrar facilmente uma informação (dado a capturar ou a consultar)?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**• A apresentação lhe parece ser homogênea de uma tela para outra?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?



**5. O que você pensa sobre os princípios de captura (entrada de dados)?**

- **O modo de captura lhe parece adaptado às suas necessidades cotidianas (uso do teclado, do mouse, de uma caneta luminosa...)?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **Os símbolos e códigos utilizados para capturar ou consultar dados são explícitos, compreensíveis?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**6. As funções ou comandos perigosos (com risco de perda de informação) lhe parecem estar protegidos?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**7. A gestão de erros é suficientemente orientada?**

- **As mensagens de erro são fáceis de compreender?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **As mensagens ajudam a encontrar a causa do erro ou do incidente?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **As mensagens permitem encontrar uma saída (solução do problema, contactar um técnico...)?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **As retomadas dos erros são possíveis, satisfatórias?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

## **8. Você pode adaptar o produto às suas necessidades pessoais?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

## **9. O que você pensa sobre a documentação em papel?**

- **Ela é clara?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **Ela é suficiente?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **Ela é eficaz para tratar um problema?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

## **10. O que você pensa sobre a documentação na tela?**

- **Ela é clara?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **Ela é suficiente?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **Ela é eficaz para tratar um problema?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

## **11. Que tipo de documentação você utiliza?**

- **Qual é sua frequência de uso da documentação em papel?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

- **Qual é sua freqüência de uso da documentação na tela?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**12. Como você avalia a facilidade de aprendizagem do software?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**13. Se um colega que conhece o trabalho tivesse que aprender a usar o software, poderia ele dominar rapidamente o funcionamento?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

**14. Como você avalia o modo de aprendizagem que lhe foi oferecido?**

- Antes da formação: 1 2 3 4 5

Por que, exemplos?

- Após a formação: 1 2 3 4 5

Por que, exemplos?

*Essas últimas questões são mais precisamente destinadas aos usuários experientes.*

*Responda circulando sim ou não.*

**15. As novas tarefas são aparentes?**

SIM – NÃO

Se sim, quais?

**16. As tarefas foram simplificadas?**

SIM – NÃO

Se sim, quais?

**17. Você constata uma grande especialização em seu trabalho (seu trabalho consiste agora no tratamento de um tipo de operação, de processo de ação pertencente a um registro mais restrito)?**

SIM – NÃO

Detalhe sua resposta:

**18. Você constata, ao contrário, um aumento de seu trabalho (seu trabalho consiste agora na intervenção mais ampla nos níveis suplementares)?**

SIM – NÃO

Detalhe sua resposta:

**19. Você constata uma nova divisão do trabalho entre você e os outros (seus colegas, sua hierarquia direta) ?**

SIM – NÃO

Detalhe sua resposta:

**ANEXO 3**







**ANEXO 4**



Florianópolis, agosto de 2003.

**Prezado Professor,**

O questionário abaixo tem por fim analisar quais os atributos de softwares comerciais para roteirização são considerados importantes para a realização da tarefa de roteirizar e programar veículos. Ao preenchê-lo, você estará contribuindo para o desenvolvimento de uma nova metodologia de avaliação de softwares interativos, a qual incorpora parâmetros de subjetividade na análise de preferência do usuário. O mesmo faz parte de um estudo que servirá de base para a tese de **Karin C. Siqueira Ramos**, doutoranda da Universidade Federal de Santa Catarina.

Este instrumento consiste em um *checklist*, onde os atributos deverão ser escolhidos de acordo com a sua relevância, segundo a opinião pessoal do respondente. Esse deve, num primeiro momento, selecionar, dentre os atributos listados, os 10 que em sua opinião são os mais essenciais para um software de roteirização. Após, o mesmo deve selecionar, em cada conjunto apresentado, os 5 atributos que considera mais importantes. Ao final da pesquisa há um espaço para sugestões onde o respondente pode adicionar algum atributo de softwares de roteirização que ele julgue importante e que não conste na listagem.

Gostaríamos de salientar que sua opinião será de grande valia na elaboração da tese de doutorado que ora propomos. Agradecemos seu empenho no preenchimento do referido questionário e tomamos a liberdade de solicitar sua resposta (via *mail*), preferivelmente, até o dia 30 de setembro.

Sinceramente,

**Karin C. Siqueira Ramos**, Msc.  
Dept<sup>o</sup>. de Engenharia de Produção e Sistemas  
Universidade Federal de Santa Catarina

**Mirian Buss Gonçalves**, Dra.  
Dept<sup>o</sup>. de Matemática  
Universidade Federal de Santa Catarina

## Checklist para avaliação dos atributos desejáveis para um software de roteirização de veículos

Primeiro caso: Gostaríamos que você selecionasse na lista abaixo os 10 (dez) atributos que você considera **essencial** e que todo bom roteirizador deve contemplar. Para selecionar os atributos, marque um "x" na coluna **SIM**.

<b>ATRIBUTO</b>	<b>SIM</b>
1. Considerar uma ou múltiplas bases	
2. Considerar diferentes tipos de veículos	
3. Considerar janelas de tempo para cada cliente	
4. Considerar tempos de carga e descarga	
5. Considerar velocidades variáveis	
6. Considerar a contratação de terceiros	
7. Considerar os limites de capacidade (peso e volume) do veículo	
8. Considerar múltiplos compartimentos por veículo	
9. Considerar a duração máxima de um roteiro	
10. Contabilizar as horas-extras dos motoristas	
11. Considerar os horários de início e término da viagem	
12. Considerar roteiros com pernoite e/ou troca de motorista	
13. Considerar a jornada de trabalho do motorista	
14. Considerar locais e/ou horários de parada fixos, por exemplo, para almoço	
15. Considerar restrições de tamanho do veículo para um cliente específico	
16. Inserir barreiras e restrições de circulação de veículos	
17. Considerar mais de um roteiro por veículo	
18. Considerar a possibilidade de coletas de retorno ( <i>backhaul</i> )	
19. Inserir e tratar informações reais de tráfego	
20. Possibilitar a mudança manual das soluções ( <i>drag and drop</i> )	
21. Disponibilizar recursos gráficos (mapas, principalmente)	
22. Geocodificar os pontos de paradas pelo endereço	
23. Calcular valores de distância e tempo reais	
24. Inferir sobre informações reais de tráfego	
25. Emitir relatório de planejamento do carregamento	
26. Emitir relatório de programação turno a turno ao motorista	
27. Resolver problemas de tamanho ilimitado	
28. Roteirizar por arcos e por nós	
29. Roteirizar em tempo real	
30. Interagir com outros softwares (rastreador, por exemplo)	

Segundo caso: Gostaríamos que agora você selecionasse, em **CADA** um dos conjuntos abaixo, os 5 (cinco) atributos que você considera **mais importantes** um bom roteirizador contemplar. Para selecionar os atributos, marque um “x” na coluna **SIM**.

### Conjunto 1 – Quanto à capacidade algorítmica

<b>ATRIBUTO</b>	<b>SIM</b>
1. Considerar janelas de tempo para cada cliente	
2. Considerar tempos de carga e descarga	
3. Considerar velocidades variáveis	
4. Considerar os horários de início e término da viagem	
5. Inserir barreiras e restrições de circulação de veículos	
6. Considerar mais de um roteiro por veículo	
7. Considerar a possibilidade de coletas de retorno ( <i>backhaul</i> )	
8. Geocodificar os pontos de paradas pelo endereço	
9. Resolver problemas de tamanho ilimitado	
10. Roteirizar por arcos e por nós	

### Conjunto 2 – Quanto às restrições consideradas

<b>ATRIBUTO</b>	<b>SIM</b>
1. Considerar uma ou múltiplas bases	
2. Considerar diferentes tipos de veículos	
3. Considerar a contratação de terceiros	
4. Considerar os limites de capacidade (peso e volume) do veículo	
5. Considerar múltiplos compartimentos por veículo	
6. Considerar a duração máxima de um roteiro	
7. Contabilizar as horas-extras dos motoristas	
8. Considerar roteiros com pernoite e/ou troca de motorista	
9. Considerar a jornada de trabalho do motorista	
10. Considerar locais e/ou horários de parada fixos, por exemplo, para almoço	
11. Considerar restrições de tamanho do veículo para um cliente específico	

**Conjunto 3 – Quanto à interface e o tratamento de dados reais**

<b>ATRIBUTO</b>	<b>SIM</b>
1. Inserir e tratar informações reais de tráfego	
2. Possibilitar a mudança manual das soluções ( <i>drag and drop</i> )	
3. Disponibilizar recursos gráficos (mapas, principalmente)	
4. Calcular valores de distância e tempo reais	
5. Inferir sobre informações reais de tráfego	
6. Emitir relatório de planejamento do carregamento	
7. Emitir relatório de programação turno a turno ao motorista	
8. Roteirizar em tempo real	
9. Interagir com outros softwares (rastreador, por exemplo)	

Há outros atributos que você julgue importante um roteirizador contemplar? Quais?

---

---

---

---

---



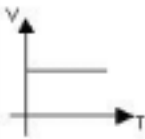



---

---


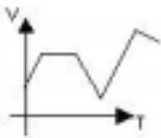

---


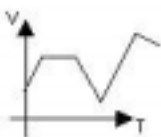


---



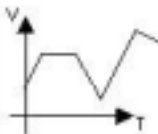



**ANEXO 5**

QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						A
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Rígidas	Não considera	Fixa	Não considera	Um único	Não considera	



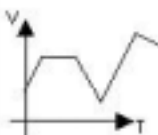



QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						B
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Rígidas	Não considera	Fixa	Considera	Múltiplos	Considera	

QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						C
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Rígidas	Considera	Variável	Não considera	Um único	Considera	



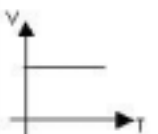



QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						D
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Rígidas	Considera	Variável	Considera	Múltiplos	Não considera	

QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						E
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Flexíveis	Não considera	Variável	Não considera	Múltiplos	Considera	



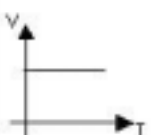



  

QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						F
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Flexíveis	Não considera	Variável	Considera	Um único	Não considera	

QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						G
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Flexíveis	Considera	Fixa	Não considera	Múltiplos	Não considera	

QUANTO À CAPACIDADE ALGORÍTMICA DO ROTEIRIZADOR						H
Janelas de tempo	Tempo de carga e descarga	Considera a velocidade	Horário de início e fim da viagem	Roteiros por veículo	Restrições de circulação de veículos	
						
Flexíveis	Considera	Fixa	Considera	Um único	Considera	



**ANEXO 6**

**QUANTO ÀS RESTRIÇÕES CONSIDERADAS**

A

Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente
				
Uma única	Único tipo	Não considera	Não considera	Não considera

**QUANTO ÀS RESTRIÇÕES CONSIDERADAS**

B

Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente
				
Uma única	Único tipo	Não considera	Considera	Considera

**QUANTO ÀS RESTRIÇÕES CONSIDERADAS**

C

Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente
				
Uma única	Vários tipos	Considera	Não considera	Considera

**QUANTO ÀS RESTRIÇÕES CONSIDERADAS**

D

Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente
				
Uma única	Vários tipos	Considera	Considera	Não considera

QUANTO ÀS RESTRIÇÕES CONSIDERADAS					E
Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente	
					
Múltiplas	Único tipo	Considera	Não considera	Considera	

QUANTO ÀS RESTRIÇÕES CONSIDERADAS					F
Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente	
					
Múltiplas	Único tipo	Considera	Considera	Não considera	

QUANTO ÀS RESTRIÇÕES CONSIDERADAS					G
Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente	
					
Múltiplas	Vários tipos	Não considera	Não considera	Não considera	






  

QUANTO ÀS RESTRIÇÕES CONSIDERADAS					H
Número de bases (origem / destino)	Tipo de veículo	Duração máxima de um roteiro	Capacidade do veículo (Ton/m <sup>3</sup> )	Tamanho do veículo por cliente	
					
Múltiplas	Vários tipos	Não considera	Considera	Considera	

**ANEXO 7**

**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS**

A

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Impossível	Impossível	Escassos	Impossível	Nenhuma

**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS**

B

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Impossível	Impossível	Escassos	Possível	Existente

**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS**

C

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Impossível	Possível	Mapas/figuras	Impossível	Existente

**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS**

D

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Impossível	Possível	Mapas/figuras	Possível	Nenhuma






**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS****E**

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Possível	Impossível	Mapas/figuras	Impossível	Existente


**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS****F**

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Possível	Impossível	Mapas/figuras	Possível	Nenhuma

**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS****G**

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Possível	Possível	Escassos	Impossível	Nenhuma

**QUANTO À INTERFACE E O TRATAMENTO DE DADOS REAIS****H**

Roteirização em tempo real	Mudança manual da solução	Recursos gráficos	Cálculo real de distância e tempo	Interação com outros softwares
				
Possível	Possível	Escassos	Possível	Existentes

**ANEXO 8**

**QUANTO AO SOFTWARE COMO UM TODO****A**

Capacidade algorítmica



Pequena  
(não aceita diferentes  
configurações)

Tratamento das restrições



Inserir poucas restrições  
(resultado pouco verídico)

Interface e tratamento de dados reais



Pouco satisfatório  
(interface ruim / não  
trabalha com dados reais)

**QUANTO AO SOFTWARE COMO UM TODO****B**

Capacidade algorítmica



Pequena  
(não aceita diferentes  
configurações)

Tratamento das restrições



Inserir várias restrições  
(resultado bastante  
semelhante ao real)

Interface e tratamento de dados reais



Satisfatório  
(boa interface / faz uso de  
dados reais)

**QUANTO AO SOFTWARE COMO UM TODO****C**

Capacidade algorítmica



Grande  
(aceita várias  
configurações da rede)

Tratamento das restrições



Inserir poucas restrições  
(resultado pouco verídico)

Interface e tratamento de dados reais



Satisfatório  
(boa interface / faz uso de  
dados reais)

**QUANTO AO SOFTWARE COMO UM TODO****D**

Capacidade algorítmica



Grande  
(aceita várias  
configurações da rede)

Tratamento das restrições



Inserir várias restrições  
(resultado bastante  
semelhante ao real)

Interface e tratamento de dados reais



Pouco satisfatório  
(interface ruim / não  
trabalha com dados reais)



**ANEXO 9**

Esta pesquisa busca dados para analisar os atributos relevantes em softwares para roteirização de veículos. Neste documento você encontrará as instruções para o preenchimento do questionário e um vocabulário que poderá lhe auxiliar na análise dos cartões que lhe serão apresentados.

**Instruções para realização da pesquisa:**

Os atributos avaliados estão divididos em três conjuntos:

- Conjunto 1 – Quanto à capacidade algorítmica do roteirizador
- Conjunto 2 – Quanto às restrições consideradas
- Conjunto 3 – Quanto à interface e o tratamento de dados reais

Para cada um desses conjuntos você encontrará, no arquivo questionário.doc, uma tabela. Nessa tabela são identificados sete blocos, cada um representado por uma cor. Cada bloco é composto por quatro cartões (os cartões estão nomeados de A até H, como pode ser visto no canto superior direito de cada cartão)

No arquivo cartões.doc você encontrará cada um desses blocos com seus respectivos cartões. Em cada um dos blocos você deve escolher aquele cartão que mais lhe satisfaz, marcando na tabela correspondente, no arquivo questionário.doc a letra associada a ele (você pode marcar o cartão escolhido com um X ao lado da letra).

Após você ter feito esse procedimento para todos os blocos dos três conjuntos, você deve ordenar, conforme sua preferência, os quatro cartões apresentados no Conjunto 4 – Quanto ao software como um todo. Da mesma forma, no arquivo questionário.doc, existe uma tabela para serem registradas essas opções.

Ao final do preenchimento do questionário, solicitamos que você envie o mesmo para o seguinte endereço eletrônico: [kramos@icablenet.com.br](mailto:kramos@icablenet.com.br).

Agradecemos seu empenho no preenchimento do referido questionário. Sua opinião será de grande valia na elaboração da tese de doutorado de Karin C. Siqueira Ramos, doutoranda da Universidade Federal de Santa Catarina.

Sinceramente,

**Karin C. Siqueira Ramos, Msc.**

Depto de Engenharia de Produção e Sistemas - UFSC

Telefone: (48) 2341914

## **Vocabulário utilizado nos cartões:**

### **Conjunto 1 – Quanto à capacidade algorítmica do roteirizador**

- Janelas de tempo: horário pré-determinado por um cliente para o recebimento de mercadorias. Se o software determina a duração dessas janelas, elas são rígidas; se o operador do software pode inserir qualquer horário, elas são então flexíveis.
- Tempo de carga e descarga: o tempo que se irá levar para carregar e/ou descarregar o veículo é ou não considerado pelo software.
- Considera a velocidade: um software pode trabalhar com uma velocidade fixa (por exemplo, a velocidade média) ou variável (considerando diferentes velocidades, dependendo da rodovia ou do horário, por exemplo).
- Horário de início e fim da viagem: se, por exemplo, a política de jornada de trabalho da minha empresa for fixa (das 8 as 18, por exemplo), é importante o software considerar esse atributo. Caso não haja nenhuma restrição na minha empresa de horário para início ou fim de uma viagem, então esse fator não é relevante.
- Roteiros por veículo: um software pode trabalhar alocando apenas um roteiro (viagem) para cada veículo por dia, ou então pode alocar a ele vários roteiros, havendo necessidade, nesse segundo caso, de um novo carregamento do veículo na origem.
- Restrições de circulação de veículos: horários onde é proibida a circulação de veículos de grande porte, rodízio de veículos, de acordo com o número da placa e vias de mão única são exemplos de restrições de circulação de veículos. Um software pode ou não considerar essas informações em sua roteirização.

### **Conjunto 2 – Quanto às restrições consideradas**

- Número de bases: por base entende-se o ponto de saída e/ou chegada dos veículos. O software pode considerar uma única ou múltiplas bases.
- Tipo de veículo: alguns softwares só trabalham com roteirização considerando apenas um único tipo de veículo (por exemplo, um caminhão do tipo X), enquanto que outros softwares consideram vários tipos de veículo (caminhões de diferentes portes, carretas, *pick-ups*, motos etc).

- Duração máxima de um roteiro: em algumas situações essa pode ser uma variável relevante, principalmente no que diz respeito às leis trabalhistas ou à política de carga horária da empresa.
- Capacidade do veículo: neste caso considera-se como informação importante o limite máximo possível de embarque em termos de peso e/ou volume. Um software pode ou não considerar essas informações em sua roteirização.
- Tamanho do veículo por cliente: alguns clientes têm restrições quanto ao tamanho do veículo que fará a entrega de sua mercadoria, principalmente pelo fato de certos veículos poderem facilitar o processo de embarque/desembarque de mercadorias. Um software pode ou não considerar essas informações em sua roteirização.

### **Conjunto 3 – Quanto à interface e o tratamento de dados reais**

- Roteirização em tempo real: imagine a seguinte situação: um caminhão está fazendo a coleta de mercadorias; na fábrica recebe-se o pedido de coleta em um cliente que está próximo deste veículo. Nesse caso, poder-se-ia inserir esse ponto de coleta na rota que esse veículo está fazendo. Em termos de software, isso pode ser possível ou não de ser feito em tempo real.
- Mudança manual da solução: Também conhecido como *drag and drop*. É a capacidade ou não de mudar a solução final dada pelo software, caso isso seja preciso (por exemplo, se for para privilegiar as exigências de algum cliente preferencial).
- Recursos gráficos: um software pode apresentar o roteiro apenas através de uma lista seqüencial ou através de pontos ligados por linhas. Se isso acontece, dizemos que seus recursos são escassos. Por outro lado, o software pode permitir a visualização da rota através de mapas e figuras explicativas.
- Cálculo real de distância e tempo: um software pode ou não ser capaz de considerar informações reais de tráfego, considerando, por exemplo, engarrafamentos e acidentes para a previsão do horário de chegada de uma mercadoria para um certo cliente.
- Interação com outros softwares: o software pode ou não ter alguma interação com outros programas computacionais, como rastreadores, WMS, GPS etc.



Esta pesquisa busca dados para analisar os atributos relevantes em softwares para roteirização de veículos. As informações aqui prestadas são confidenciais e serão utilizadas em pesquisa científica. Sua opinião será de grande valia na elaboração de uma nova metodologia de avaliação de software, a qual incorpora parâmetros de subjetividade na análise da preferência do usuário.

Idade			
Até 25 anos	<input type="checkbox"/>	Até 35 anos	Até 45 anos <input type="checkbox"/>
Acima de 45			
Escolaridade			
1º grau	<input type="checkbox"/>	2º grau	3º grau
Pós-graduado			
Possui acesso ao computador:			
Em casa	<input type="checkbox"/>	No trabalho	Em ambos
Você utiliza softwares roteirizadores de veículo?			
Sim <input type="checkbox"/>		Não <input type="checkbox"/>	
Se sua resposta foi SIM, com que frequência isso ocorre:			
Diária <input type="checkbox"/>		1 ou 2 vezes por semana	
1 ou 2 vezes por mês <input type="checkbox"/>		Com menor frequência	
Se sua resposta foi NÃO, por que isso ocorre:			
Não faço uso de roteirizadores porque a empresa/instituição não os possui			<input type="checkbox"/>
Não faço uso de roteirizadores porque não conheço essa ferramenta			<input type="checkbox"/>
Não faço uso de roteirizadores porque não sei usá-los			<input type="checkbox"/>
Não faço uso de roteirizadores porque não os acho relevantes			<input type="checkbox"/>
Outros motivos			<input type="checkbox"/>

### Quanto à Capacidade Algorítmica do Roteirizador (Conjunto 1)

(marque em cada bloco a alternativa que mais lhe satisfaz)

Bloco 1 (azul)	A	B	C	D
Bloco 2 (laranja)	A	B	E	F
Bloco 3 (verde)	A	B	G	H
Bloco 4 (vermelho)	A	C	E	G
Bloco 5 (cinza)	A	C	F	H
Bloco 6 (marrom)	A	D	E	H
Bloco 7 (rosa)	A	D	F	G

**Quanto às Restrições Consideradas (Conjunto 2)**

(marque em cada bloco a alternativa que mais lhe satisfaz)

Bloco 1 (azul)	A	B	C	D
Bloco 2 (laranja)	A	B	E	F
Bloco 3 (verde)	A	B	G	H
Bloco 4 (vermelho)	A	C	E	G
Bloco 5 (cinza)	A	C	F	H
Bloco 6 (marrom)	A	D	E	H
Bloco 7 (rosa)	A	D	F	G

**Quanto à Interface e o Tratamento de Dados Reais (Conjunto 3)**

(marque em cada bloco a alternativa que mais lhe satisfaz)

Bloco 1 (azul)	A	B	C	D
Bloco 2 (laranja)	A	B	E	F
Bloco 3 (verde)	A	B	G	H
Bloco 4 (vermelho)	A	C	E	G
Bloco 5 (cinza)	A	C	F	H
Bloco 6 (marrom)	A	D	E	H
Bloco 7 (rosa)	A	D	F	G

**Quanto ao software como um todo (Conjunto 4):**

(ordene os 4 cartões conforme sua preferência, colocando na coluna da direita a letra correspondente ao cartão: A, B, C ou D)

1ª opção	
2ª opção	
3ª opção	
4ª opção	

(Para uso do pesquisador)

**Entrevista n°:**

**ANEXO 10**

## Conjunto 1 – Estratificado

### Empresas

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (7) Iterações \*\*\*  
 Eficiência = 0,8571 \*\*\* QMR = 1,3663

Atributo	Coef.	Erro	Teste t	IC. (t=2,5%)
Janelas tempo	0,6166	0,2697	2,2859	[0,077 ; 1,156]
Tempo carga e d	1,3155	0,3060	4,2993	[0,704 ; 1,927]
Velocidade	0,6934	0,2867	2,4184	[0,120 ; 1,267]
Hora início e f	1,0894	0,2776	3,9237	[0,534 ; 1,645]
Roteiros por ve	1,0137	0,3173	3,1953	[0,379 ; 1,648]
Restrições circ	1,2882	0,3248	3,9662	[0,639 ; 1,938]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 98	Número de Casos = 98
F(Betas_0) = -135,8568	F(Betas_1) = -93,0068
F(Blocos_0) = -190,6992	F(Blocos_1) = -193,9117
LR (-2[F(0)-F(B)])= 85,7002	LR (Blocos) = -6,4250
Rho = 0,3154	Rho (Ajt) = 0,2712
Rho (Blc) = -0,0168	

### Pesquisadores

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (7) Iterações \*\*\*  
 Eficiência = 0,8571 \*\*\* QMR = 2,6695

Atributo	Coef.	Erro	Teste t	IC. (t=2,5%)
Janelas tempo	0,8744	0,3177	2,7523	[0,239 ; 1,510]
Tempo carga e d	1,1161	0,3386	3,2965	[0,439 ; 1,793]
Velocidade	1,5219	0,4543	3,3497	[0,613 ; 2,431]
Hora início e f	1,5810	0,4422	3,5750	[0,697 ; 2,465]
Roteiros por ve	0,7974	0,3338	2,3893	[0,130 ; 1,465]
Restrições circ	1,7592	0,4563	3,8553	[0,847 ; 2,672]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 77	Número de Casos = 77
F(Betas_0) = -106,7447	F(Betas_1) = -68,3623
F(Blocos_0) = -149,8351	F(Blocos_1) = -152,4730
LR (-2[F(0)-F(B)])= 76,7648	LR (Blocos) = -5,2758
Rho = 0,3596	Rho (Ajt) = 0,3034
Rho (Blc) = -0,0176	



## Conjunto 2 – Estratificado

### Empresas

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (6) Iterações \*\*\*  
 Eficiência = 0,8571 \*\*\* QMR = 1,0681

Atributo	Coef.	Erro	Teste t	IC. (t=2,5%)
Número de bases	0,3809	0,3055	1,2468	[-0,230 ; 0,992]
Tipo de veículo	1,4395	0,3055	4,7123	[0,829 ; 2,050]
Duração máxima	1,3967	0,3643	3,8339	[0,668 ; 2,125]
Capacidade veíc	1,4039	0,3232	4,3442	[0,758 ; 2,050]
Tamanho veículo	0,7111	0,3232	2,2006	[0,065 ; 1,357]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 98      Número de Casos = 98  
 F(Betas\_0) = -135,8568      F(Betas\_1) = -78,3311  
 F(Blocos\_0) = -190,6992      F(Blocos\_1) = -197,5462  
 LR (-2[F(0)-F(B)]) = 115,0515      LR (Blocos) = -13,6941  
 Rho = 0,4234      Rho (Ajt) = 0,3866  
 Rho (Blc) = -0,0359

### Pesquisadores

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (4) Iterações \*\*\*  
 Eficiência = 0,8571 \*\*\* QMR = 1,7524

Atributo	Coef.	Erro	Teste t	IC. (t=2,5%)
Número de bases	1,2822	0,4364	2,9381	[0,409 ; 2,155]
Tipo de veículo	1,5118	0,4364	3,4640	[0,639 ; 2,385]
Duração máxima	1,4986	0,4616	3,2466	[0,575 ; 2,422]
Capacidade veíc	1,3204	0,3194	4,1339	[0,682 ; 1,959]
Tamanho veículo	0,1866	0,3194	0,5842	[-0,452 ; 0,825]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 77      Número de Casos = 77  
 F(Betas\_0) = -106,7447      F(Betas\_1) = -56,0334  
 F(Blocos\_0) = -149,8351      F(Blocos\_1) = -153,8759  
 LR (-2[F(0)-F(B)]) = 101,4225      LR (Blocos) = -8,0816  
 Rho = 0,4751      Rho (Ajt) = 0,4282  
 Rho (Blc) = -0,0270

### Conjunto 3 – Estratificado

#### Empresas

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (6) Iterações \*\*\*  
 Eficiência = 0,8571 \*\*\* QMR = 0,7770

Atributo	Coef.	Erro	Teste t	IC. (t=2,5%)
Roteirização te	0,7002	0,2591	2,7023	[0,182 ; 1,218]
Mudança manual	0,9233	0,2591	3,5636	[0,405 ; 1,442]
Recursos gráfico	1,0503	0,3161	3,3224	[0,418 ; 1,683]
Cálculo real	1,1634	0,2974	3,9119	[0,569 ; 1,758]
Interação softw	1,0886	0,2974	3,6604	[0,494 ; 1,683]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 98      Número de Casos = 98  
 F(Betas\_0) = -135,8568      F(Betas\_1) = -92,4460  
 F(Blocos\_0) = -190,6992      F(Blocos\_1) = -194,6028  
 LR (-2[F(0)-F(B)]) = 86,8217      LR (Blocos) = -7,8072  
 Rho = 0,3195      Rho (Ajt) = 0,2827  
 Rho (Blc) = -0,0205

#### Pesquisadores

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (6) Iterações \*\*\*  
 Eficiência = 0,8571 \*\*\* QMR = 2,0017

Atributo	Coef.	Erro	Teste t	IC. (t=2,5%)
Roteirização te	1,3822	0,4229	3,2681	[0,536 ; 2,228]
Mudança manual	1,3822	0,4229	3,2681	[0,536 ; 2,228]
Recursos gráfico	1,6943	0,4837	3,5030	[0,727 ; 2,662]
Cálculo real	1,2817	0,3651	3,5103	[0,551 ; 2,012]
Interação softw	1,0194	0,3651	2,7918	[0,289 ; 1,750]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 77      Número de Casos = 77  
 F(Betas\_0) = -106,7447      F(Betas\_1) = -64,0964  
 F(Blocos\_0) = -149,8351      F(Blocos\_1) = -153,2833  
 LR (-2[F(0)-F(B)]) = 85,2966      LR (Blocos) = -6,8965  
 Rho = 0,3995      Rho (Ajt) = 0,3527  
 Rho (Blc) = -0,0230

### Conjunto 4 – Estratificado

#### Empresas

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (4) Iterações \*\*\*  
 Eficiência = 1,0000 \*\*\* QMR = 1,9749

Atributo	Coef.	Erro	Teste t	IC.(t=2,5%)
Capacidade algo	1,6456	0,5618	2,9293	[0,522 ; 2,769]
Tratamento das	1,7173	0,5486	3,1306	[0,620 ; 2,814]
Interface e dad	1,2524	0,5405	2,3170	[0,171 ; 2,333]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 14

Número de Casos = 42

F(Betas\_0) = -44,4928

F(Betas\_1) = -25,7584

LR (-2[F(0)-F(B)])= 37,4687

Rho = 0,4211

Rho (Ajt) = 0,3536

#### Pesquisadores

\*\*\* LOGIT MULTINOMIAL COM PPROBABILIDADE CONDICIONAL \*\*\*  
 \*\*\* Método de Newton-Raphson - Ponto Máximo com (4) Iterações \*\*\*  
 Eficiência = 1,0000 \*\*\* QMR = 3,2370

Atributo	Coef.	Erro	Teste t	IC.(t=2,5%)
Capacidade algo	1,0931	0,6663	1,6405	[-0,240 ; 2,426]
Tratamento das	2,4884	0,7441	3,3443	[1,000 ; 3,977]
Interface e dad	1,7515	0,6737	2,5998	[0,404 ; 3,099]

Obs.: A eficiência já está incluída na Variância.

Transformação: DADOS NÃO TRANSFORMADOS

Número de Entrevistas = 11

Número de Casos = 33

F(Betas\_0) = -34,9586

F(Betas\_1) = -17,7423

LR (-2[F(0)-F(B)])= 34,4325

Rho = 0,4925

Rho (Ajt) = 0,4067