

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

PROPOSTA DE AVALIAÇÃO DA QUALIDADE DE
PRODUTOS DE SOFTWARE UTILIZANDO
A NORMA ISO/IEC 9126

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À
UNIVERSIDADE REGIONAL DE BLUMENAU PARA OBTENÇÃO
DOS CRÉDITOS DE DISCIPLINA COM NOME EQUIVALENTE NO
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

MIRIAN MIRDES STORCH

BLUMENAU, JUNHO DE 2000

2000/1-53

UNIVERSIDADE REGIONAL DE BLUMENAU
CENTRO DE CIÊNCIAS EXATAS E NATURAIS
CURSO DE CIÊNCIAS DA COMPUTAÇÃO
(Bacharelado)

PROPOSTA DE AVALIAÇÃO DA QUALIDADE DE
PRODUTOS DE SOFTWARE UTILIZANDO
A NORMA ISO/IEC 9126

TRABALHO DE CONCLUSÃO DE CURSO SUBMETIDO À
UNIVERSIDADE REGIONAL DE BLUMENAU PARA OBTENÇÃO
DOS CRÉDITOS DE DISCIPLINA COM NOME EQUIVALENTE NO
CURSO DE CIÊNCIAS DA COMPUTAÇÃO – BACHARELADO

MIRIAN MIRDES STORCH

AVALIAÇÃO DA QUALIDADE DE PRODUTOS DE SOFTWARE UTILIZANDO A NORMA ISO/IEC 9126

MIRIAN MIRDES STORCH

ESTE RELATÓRIO, DO TRABALHO DE CONCLUSÃO DE CURSO, FOI JULGADO ADEQUADO PARA OBTENÇÃO DOS CRÉDITOS DA DISCIPLINA DE TRABALHO DE CONCLUSÃO DE CURSO OBRIGATÓRIO PARA OBTENÇÃO DO TÍTULO DE :

BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO

Prof. José Roque Voltoline da Silva
COORDENADOR DO TCC

Prof. Everaldo Artur Grahl – Orientador

BANCA EXAMINADORA

Prof. Everaldo Artur Grahl

Prof. Wilson Carli

Prof. Carlos Eduardo Negrão Bizzotto

DEDICATÓRIA

“O homem é aquilo em que acredita. Qualquer que seja sua situação atual, você poderá cair, permanecer ou subir de acordo com seus pensamentos, sua visão, seus ideais. Você se tornará tão pequeno como seus desejos repressores ou tão dependente quanto sua aspiração dominante. Da natureza de nossos pensamento depende da fortaleza de nosso corpo, o vigor de nossa inteligência, o êxito de nossos negócios”.

Márcio Kuhne.

Á

Rodrigo Ferracin de Souza

Nos caminhos que vivemos podemos encontrar muitas dificuldades, mas em meio estas dificuldades é que surgem as oportunidades. Oportunidades de conhecer seres humanos incríveis. É bom saber, que ao lado de cada obstáculo existe alguém torcendo por nós e querendo nosso sucesso. Alguém como você.

AGRADECIMENTOS

A Deus pela energia positiva, que tem e me dado forças para vencer esta etapa tão importante de minha vida.

A meus pais, que não foram apenas pais, mas amigos e companheiros, compartilhando meus ideais e contribuindo de uma forma especial para que os meus objetivos fossem alcançados, pela lição de amor que me ensinaram durante toda a minha vida.

Aos colegas de curso, que durante todo esse período acompanharam-me, apontando sempre uma atitude de vontade e persistência aos trabalhos.

Agradeço a toda equipe de coordenadores e professores do curso, que sempre nos proporcionaram todo o apoio e com o máximo de empenho a cada disciplina para o seu melhor desempenho.

Aos professores integrantes da banca pela valorização que suas contribuições prestaram ao presente trabalho.

Ao professor Everaldo Artur Grahl, orientador deste trabalho, pelo apoio técnico e profissional efetuado durante toda a elaboração e desenvolvimento da monografia.

A todos que de alguma forma, estiveram presentes em minha vida, pela amizade, carinho, companheirismo, que a mim se ligaram através de um vínculo de experiência comum.

Com carinho,

Mirian Mirdes Storch.

SUMÁRIO

	DEDICATÓRIA	iii
	AGRADECIMENTOS	v
	SUMÁRIO	vi
	LISTA DE FIGURAS	ix
	LISTA DE TABELAS	x
	LISTA DE ABREVIATURAS E SIGLAS	xi
	RESUMO	xii
	ABSTRACT	xiii
1.	INTRODUÇÃO	1
1.1	ORIGEM	1
1.2	OBJETIVOS	2
1.3	ORGANIZAÇÃO	3
2.	QUALIDADE	4
2.1	INTRODUÇÃO	4
2.2	CONCEITO DE QUALIDADE	4
2.3	HISTÓRICO E EVOLUÇÃO	5
2.4	CONSIDERAÇÕES SOBRE A QUALIDADE	5
2.5	QUALIDADE DE SOFTWARE	6
2.5.1	DEFINIÇÕES	6
2.6	FATORES EXPLÍCITOS E IMPLÍCITOS	8
2.7	VISÕES DA QUALIDADE DE SOFTWARE	9
2.7.1	VISÃO DO USUÁRIO	9

2.7.2	VISÃO DA EQUIPE DE DESENVOLVIMENTO	10
2.7.3	VISÃO DO GERENTE	10
3.	NORMA ISO/IEC 9126	11
3.1	OBJETIVOS DA NORMA	11
3.2	MÉTODO DE AVALIAÇÃO	11
3.2.1	DEFINIÇÃO DE REQUISITOS DA QUALIDADE	12
3.2.2	ESPECIFICAÇÃO E PREPARAÇÃO DA AVALIAÇÃO	13
3.2.3	EXECUÇÃO DA AVALIAÇÃO	14
3.3	QUALIDADE DE PRODUTO DE SOFTWARE	15
3.3.1	CARACTERÍSTICAS DA QUALIDADE E MÉTRICA- ISO/IEC 9126-1	16
3.3.1.2	FUNCIONALIDADE	16
3.3.1.3	CONFIABILIDADE	17
3.3.1.4	USABILIDADE	18
3.3.1.5	EFICIÊNCIA	18
3.3.1.6	MANUTENIBILIDADE	19
3.3.1.7	PORTABILIDADE	19
4.	PROPOSTA DE MENSURAÇÃO DA AVALIAÇÃO	
	DA QUALIDADE DE SOFTWARE	21
4.1	FUNCIONALIDADE	21
4.2	CONFIABILIDADE	23
4.3	USABILIDADE	26
4.4	EFICIÊNCIA	28
4.5	MANUTENIBILIDADE	29
4.6	PORTABILIDADE	31
5.	DESENVOLVIMENTO DO PROTÓTIPO	33

5.1	INTRODUÇÃO	33
5.2	AMBIENTE DE DESENVOLVIMENTO	33
5.3	SYSTEM ARCHITECT.....	33
5.4	DIAGRAMA DE FLUXO DE DADOS	34
5.5	MODELO ENTIDADE RELACIONAMENTO	37
5.6	DICIONÁRIO DE DADOS	38
5.7	DIAGRAMA HIRÁRQUICO FUNCIONAL	41
5.8	DESCRIÇÃO DO PROTÓTIPO	42
5.9	FUNCIONAMENTO DO SOFTWARE – DESCRIÇÃO DAS TELAS	46
5.10	ANÁLISE DO PROCESSO DE AVALIAÇÃO E RESULTADOS	56
6.	CONCLUSÃO	58
ANEXO A – QUESTIONÁRIO DE AVALIAÇÃO DAS CARACTERÍSTICAS		
	DA QUALIDADE DE PRODUTOS DE SOFTWARE	59
ANEXO B – EXEMPLO DE AVALIAÇÃO DE PRODUTOS DE		
	SOFTWARE	69
	GLOSSÁRIO	79
	REFERÊNCIAS BIBLIOGRÁFICAS	80

LISTA DE FIGURAS

1	Modelo de Processo de Avaliação	12
2	Diagrama da Estrutura da Característica Funcionalidade	22
3	Diagrama da Estrutura da Característica Confiabilidade	24
4	Diagrama da Estrutura da Característica Usabilidade	26
5	Diagrama da Estrutura da Característica Eficiência	28
6	Diagrama da Estrutura da Característica Manutenibilidade	29
7	Diagrama da Estrutura da Característica Portabilidade	31
8	Diagrama de Contexto	34
9	Diagrama de Fluxo de Dados Nível 0	36
10	Modelo Entidade Relacionamento	37
11	Diagrama Hierárquico Funcional	42
12	Tela Principal do Protótipo	47
13	Tela de Cadastro de Avaliador	47
14	Tela de Cadastro de Software	48
15	Tela de Cadastro de Categoria	48
16	Tela de Cadastro de Característica	49
17	Tela de Cadastro de Graus	49
18	Tela de Cadastro de Pesos	50
19	Tela de Configurar Avaliação	50
20	Tela de Seleção das características	51
21	Tela de Itens Avaliados	52
22	Tela de Grau de Importância.....	52

23	Tela de Resultados	53
24	Tela de Consulta de Avaliadores	53
25	Tela Página Inicial do Relatório de Resultados	54
26	Tela Página Final do Relatório de Resultados	54
27	Tela de Relatório das Avaliações Realizadas	55
28	Tela de Classificação do Software	55

LISTA DE TABELAS

1	Fatores Explícitos	8
2	Fatores Implícitos	9
3	Questionário de avaliação das características da qualidade de softwares	59
4	Exemplo de Avaliação de Produto de Software	69

LISTA DE ABREVIATURA E SIGLAS

ABNT	- Associação Brasileira de Normas Técnicas
CASE	- <i>Computer Aided Systems Enginnering</i>
DFD	- Diagrama de Fluxo de Dados
DHF	- Diagrama Hierárquico Funcional
MER	- Entidade/Relacionamento
IEC	- (<i>International Eletrotechnical Committe</i>) – <i>Comitê Eletrotécnico Internacional</i>
ISO	- (<i>International Organization Standardization</i>) – Organização Internacional de Normalização
TAQS	- Tecnologia para Avaliação da qualidade de Software

RESUMO

Este trabalho visa apresentar um estudo sobre a norma ISO/IEC 9126 (International Organization for Standardization / Internacional Eletronical Commission). Além disso, o trabalho propõe uma forma para a avaliação da Qualidade de Produtos de Software baseado na norma ISO/IEC 9126. Com o objetivo de tornar a avaliação menos subjetiva, faz-se um detalhamento das características e subcaracterísticas desta norma. Um protótipo para avaliação da qualidade de software foi implementado para apoiar o avaliador durante o processo de avaliação.

ABSTRACT

This project presents a study about the ISO/IEC 9126 (International Organization for Standardization/International Electrotechnical Commission) standard. Besides, this project proposes a way to evaluate the Quality of Software Products, based on ISO/IEC 9126 standard. Purposing to change the evaluation less subjective, it's necessary to detail the characteristics and sub-characteristics from this standard. A prototype for software quality evaluation was implemented to support the evaluator during the process of evaluation.

1 INTRODUÇÃO

1.1 ORIGEM

Atualmente, a melhoria da qualidade do software tornou-se cada vez mais comum nas organizações devido à necessidade de obtenção de melhores resultados em todas as fases do ciclo de vida do software.

Conforme [BAR1997], qualidade é a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer às necessidades explícitas e implícitas.

Segundo [GIL1995], “a qualidade é difícil de ser definida, impossível de ser medida e fácil de se reconhecer. A qualidade é geralmente transparente quando presente, mas a sua ausência é facilmente percebida”. Algumas considerações podem ser feitas sobre a qualidade:

- a) qualidade não é absoluta: ela representa coisas diferentes em diferentes situações. A qualidade não pode ser medida sob escala qualificável da mesma maneira que propriedades físicas como temperatura ou tamanho;
- b) qualidade é multidimensional (ela tem muitos fatores contribuintes): ela não é facilmente sumarizada em um meio simples e quantitativo. Alguns aspectos da qualidade podem ser medidos, com maior ou menor facilidade, no entanto, o critério mais facilmente medido pode não ser necessariamente o mais importante. A aceitação de um produto pode depender de critérios que muitas vezes são difíceis de definir;
- c) qualidade é objeto de restrições: a qualidade muitas vezes não pode ser separada do custo, entendendo-se custos como pessoas, tempo e ferramentas;
- d) qualidade envolve compromissos e aceitações: quando a qualidade é restringida e compromissos são requeridos, alguns critérios podem ser sacrificados mais aceitavelmente que outros, por exemplo, conforto pode ser sacrificado no lugar de segurança;
- e) critérios de qualidade não são independentes, mas integram um com outro.

Com o objetivo de padronizar e aprimorar a avaliação da qualidade de produtos de software, surge a Norma ISO/IEC 9126 desenvolvida em conjunto com a ISO e IEC.

A norma ISO/IEC 9126, equivalente a Norma Brasileira NBR 13596, define também seis grandes grupos de características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade.

Segundo a norma ISO/IEC 9126, a definição para avaliação de produto de software refere-se à medição de quanto o produto de software, em uso, atende às necessidades implícitas e explícitas. Esta definição leva às seguintes questões:

- a) como definir os requisitos da qualidade do produto: conjunto de necessidades explícitas e implícitas;
- b) como medir a qualidade em uso;
- c) como relacionar as visões da qualidade de produto e processo;

Uma das principais dificuldades na utilização da norma ISO/IEC 9126 é a especificação do método de avaliação, principalmente a definição das métricas aplicadas. Atualmente existem poucas métricas de aceitação geral para as características descritas na norma, o que permite grupos ou organizações de normalização estabelecer os seus próprios modelos de processos de avaliação e métodos para a criação e validação de métricas relacionadas com estas características. Torna-se necessário estabelecer níveis de pontuação e critérios específicos para a aplicação das características com propósito de definição e avaliação.

1.2 OBJETIVOS

O principal objetivo do trabalho é propor um roteiro de avaliação da qualidade de produtos de software baseado na norma ISO/IEC 9126.

Os objetivos específicos são:

- a) estudar os procedimentos de avaliação da norma ISO/IEC 9126;
- b) elaborar um questionário de avaliação da qualidade para sistemas de qualquer área;
- c) desenvolver um protótipo para auxiliar na proposta de avaliação da qualidade de produtos de software.

1.3 ORGANIZAÇÃO

A seguir serão descritos brevemente cada capítulo do trabalho.

O primeiro capítulo contém a introdução do trabalho, a descrição dos objetivos e as razões pelas quais o trabalho foi realizado. Descreve-se também a metodologia adotada para o desenvolvimento do trabalho e a delimitação da abrangência do mesmo.

O segundo capítulo aborda a qualidade nas organizações, descreve a preocupação com a qualidade numa perspectiva histórica. Aborda a questão da qualidade direcionada a área de software e apresentando as definições existentes para a qualidade de software.

O terceiro capítulo descreve detalhadamente a ISO/IEC 9126, os objetivos, definições e método de avaliação, incluindo as características de avaliação do modelo.

O quarto capítulo contém a descrição da proposta de detalhamento da ISO/IEC 9126, incluindo figuras para auxiliar na compreensão do modelo.

O quinto capítulo contém a descrição do ambiente de desenvolvimento do protótipo, descrevendo a ferramenta utilizada e a especificação do protótipo implementado, o seu funcionamento, incluindo sua documentação e exemplos de telas.

O sexto capítulo contém as conclusões do trabalho realizado e sugestões para melhoramentos futuros.

2 QUALIDADE

2.1 INTRODUÇÃO

O processo de globalização da economia, tem exposto as empresas a uma concorrência mais acirrada. Por este motivo, a simples manutenção do mercado de atuação da empresa, tem sido uma tarefa bastante complexa.

O caminho encontrado por um grande número de empresas é a diferenciação de seus produtos e serviços pela qualidade. Isto tem auxiliado estas empresas não só na manutenção de seus clientes, mas também na ampliação de sua participação no mercado.

A qualidade, portanto, tem sido a responsável pelo aumento no grau de competitividade de inúmeras empresas. No entanto, apesar das vantagens associadas a qualidade existe um grande caminho a percorrer no sentido de conscientizar as empresas nessas vantagens e das ações que devem ser executadas para alcançar a qualidade.

2.2 CONCEITO DE QUALIDADE

Várias definições são dadas ao termo “Qualidade”. Lista-se a seguir um conjunto de definições citadas por [PUR95]:

- a) qualidade é a satisfação do cliente;
- b) qualidade significa:
 - conformidade aos requisitos especificados;
 - capacidade de uso;
 - valor monetário;
 - defeito zero;
 - eficiência e produtividade;
 - investimento mais lucrativo;
 - entrega a tempo;
 - credibilidade;
 - valor agregado aos produtos e serviços;
 - constância de propósitos;
 - excelência inata;
 - garantia de confiança.

Segundo [NOB1994], “Qualidade é a composição total das características de marketing, engenharia, fabricação e manutenção de um produto ou serviço, através das quais o mesmo produto ou serviço, atenderá às expectativas do cliente” ou “Qualidade ou serviço de qualidade é aquele que atende perfeitamente, de forma confiável, de forma acessível, de forma segura e no tempo certo, as necessidades do cliente”.

Dentro do contexto deste trabalho a qualidade será considerada conforme a definição da ISO/IEC 9126, ou seja, como o conjunto de características de uma entidade que se baseia na habilidade de satisfazer necessidades implícitas e explícitas.

2.3 HISTÓRICO E EVOLUÇÃO

Dentro da infra-estrutura organizacional das cidades desenvolvidas, os anos 90 constituem o que pode ser chamado de era da Qualidade e Produtividade.

“A qualidade não foi uma questão a poucas décadas atrás porque os produtores de manufatura localizavam-se em pequenas cidades e tudo o que produziam era consumido e utilizado internamente e o restante era vendido para cidades vizinhas que não produziam aquele produto” [PUR1995].

Nesta época a preocupação era unicamente com a qualidade do produto, a qual estava relacionada unicamente com as expectativas do produtor. Atualmente a qualidade é tratada de forma mais abrangente, envolvendo desde os fornecedores até os clientes finais. Além disso, existe a preocupação tanto com a qualidade do produto quanto a qualidade dos processos executados dentro da empresa. O mercado hoje, com o crescimento das indústrias está muito mais competitivo, e a qualidade surge como fator diferenciador que proporcionará vantagens competitivas aos que a detiverem e o fracasso daqueles que não estiverem preparados.

2.4 CONSIDERAÇÕES SOBRE A QUALIDADE

Como forma de sistematizar o conceito de qualidade, [NOB1994] apresenta oito dimensões sob as quais a qualidade pode ser definida:

- a) desempenho: referindo-se às características operacionais básicas de um produto;

- b) característica funcionais básicas que suplementam o funcionamento básico de um produto;
- c) confiabilidade: probabilidade de mau funcionamento de um produto;
- d) conformidade: grau em que o projeto e as características de um produto estão de acordo com as especificações;
- e) durabilidade: tempo de uso de um produto antes de se deteriorar fisicamente;
- f) atendimento: rapidez, cortesia e facilidade na execução de reparos;
- g) estética: aspectos perceptíveis pelos cinco sentidos humanos;
- h) a qualidade presente no produto que é percebida pelo cliente.

Segundo estas definições nota-se que a qualidade pode ser percebida sob ângulos bem distintos, da mesma forma o grau de qualidade de um produto pode variar dependendo do ângulo considerado. O processo de seleção de um produto dificilmente envolve todos os aspectos anteriormente citados, muitos consumidores atentam apenas para um ou outro detalhe como preço e aparência por exemplo, e deixam de considerar outros aspectos importantes. Desta maneira é importante que quando da realização de uma avaliação o produto seja valido contra todos os aspectos para que os resultados reflitam a realidade.

2.5 QUALIDADE DE SOFTWARE

2.5.1 DEFINIÇÕES

Os diversos ramos de atividade vem investindo expressivamente na implantação de sistemas de qualidade. Na indústria, já está bastante dessiminada a cultura da qualidade. A área de serviços está em um estágio anterior com relação a indústria, no entanto, os esforços que vem sendo realizados, tem trazido excelentes resultados para o setor.

Segundo [ROC1995], para que um software possa competir no mercado o projeto necessita ter uma relação custo benefício adequada e os produtos precisam ser de alta qualidade. Além disso, a melhoria na qualidade de software é condição essencial para que a empresa possa conquistar novos mercados.

Ao contrário dos setores citados, a área de software ainda está no início da difusão dos conceitos de qualidade. Por isso torna-se urgente a aplicação da filosofia da qualidade no desenvolvimento de software.

Na área de software não existe uma definição universal para a qualidade. Cada autor a define de maneira diferente e segundo critérios próprios. Isto gera dificuldades na compreensão do termo.

Segundo [CAM1992], “*um produto ou serviço de qualidade é aquele que atende perfeitamente, de forma confortável, acessível, segura e no tempo certo às necessidades do cliente*”. Esta definição pode se entendida como:

- a) projeto perfeito: que atende perfeitamente;
- b) sem defeitos: de forma confiável;
- c) baixo custo: de forma acessível;
- d) segurança do cliente: no tempo certo;

Segundo [MIL1994], qualidade de software é:

- a) ausência de defeitos;
- b) atendimento às especificações;
- c) atendimento à ISO 9000, que não apenas satisfaz às exigências, mas também é implementado a tempo e de acordo com o orçamento.

[SHI1992] define “*software de qualidade como aquele que: cumpre seus objetivos, é gerenciável, é passível de manutenção e tem longa duração e é passível de aprendizado*” .

Segundo [WEI1994], “*a qualidade é a conformidade às exigências de alguma pessoa*”. Para pessoas diferentes, um mesmo produto tem qualidade diferente.

“*Um software de qualidade deve encantar o consumidor, e não apenas funcionar direito e não ter erros*”. Bill Gates, Microsoft [WEI1994].

Entretanto, avaliações de um mesmo produto podem ser conduzidas com diferentes especificações de avaliação, levando por isso a diferentes resultados [ISO1993].

2.6 FATORES EXPLÍCITOS E IMPLÍCITOS

Segundo [FER1995], para se definir qualidade de software, é necessário entender os fatores críticos, que podem ser divididos em fatores explícitos e implícitos da qualidade.

Os fatores explícitos são externados pelo cliente. É o cliente quem define a qualidade, como a qualidade dos projetos/processos, qualidade do produto final, manutenções corretivas, adaptativas e introdução de melhorias no produto, como mostra a tabela a seguir:

TABELA 1 – FATORES EXPLÍCITOS

Fatores	Descrição
Prazo do projeto	Prazo desejado pelo cliente ou estabelecido em termos contratuais
Informação sobre progresso	Informações sobre o progresso do projeto disponibilizadas para o cliente, sua periodicidade
Atendimento funcional	Extensão em que o sistema satisfaz suas especificações e atinge os objetivos do usuário
Confiabilidade	Extensão em que o sistema desempenha suas funções requeridas com precisão
Integridade	Extensão em que o acesso ao sistema ou dados do mesmo por pessoas não autorizadas pode ser controlado
Usabilidade	Esforço requerido para aprender a operar, preparar entradas e interpretar saídas do sistema
Retornos sobre investimentos	Benefícios econômicos obtidos pelo cliente através do sistema
Tempos de atendimento	Tempos de atendimento para manutenções no produto requerido pelo cliente

Fonte: [FER1995]

Os fatores implícitos dizem respeito aos fatores da qualidade do software, que são percebidos pelos desenvolvedores, que atendem aos fatores explícitos e principalmente à produção econômica do software, como mostra a tabela a seguir:

TABELA 2 – FATORES IMPLÍCITOS

Fatores	Descrição
Manutenibilidade	Esforço requerido para localizar e remover um defeito de um módulo

	ou programa do sistema
Testabilidade	Esforço requerido para testar um programa ou módulo
Flexibilidade	Esforço necessário para modificar um programa ou módulo
Portabilidade	Esforço requerido para transferir um programa ou sistema como um todo de uma plataforma para outra
Reusabilidade	Extensão em que o programa pode ser usado em outras aplicações
Interoperabilidade	Esforço requerido para interagir/integrar sistemas entre si

Fonte: [FER1995]

2.7 VISÕES DA QUALIDADE DE SOFTWARE

Segundo a Norma ISO/IEC 9126 [ISO1996], não existe um sistema de classificação de software amplamente aceito, existem algumas classes de software que são amplamente aceitas. Deve-se atribuir a importância de cada característica da qualidade de acordo com a classe do software. A característica confiabilidade, por exemplo, é mais importante para software de missão crítica, a característica eficiência é mais importante para software de sistema em tempo real, a usabilidade é mais importante para software interativo em relação ao usuário final. A importância de cada característica da qualidade varia dependendo do ponto de vista considerado.

2.7.1 VISÃO DO USUÁRIO

Os usuários estão interessados principalmente no uso do software, no seu desempenho e nos efeitos do uso do software.

As questões do usuário pode incluir:

- a) quão confiável é o software;
- b) quão eficiente é o software;
- c) o software é fácil de usar;
- d) é possível transferir de um ambiente para outro.

2.7.2 VISÃO DA EQUIPE DE DESENVOLVIMENTO

A equipe de desenvolvimento é responsável pela produção de software que satisfaça aos requisitos de qualidade, ela está interessada tanto na qualidade dos produtos intermediários como na qualidade de produtos finais. Para a avaliação da qualidade dos produtos intermediários em cada fase do ciclo de desenvolvimento, as equipes de desenvolvimento precisam usar métricas diferentes para as mesmas características, porque as mesmas métricas não são aplicáveis em todo o ciclo de vida.

A visão das equipes de desenvolvimento também podem incorporar a visão de características da qualidade requerida por aqueles que fazem manutenção do software.

2.7.3 VISÃO DO GERENTE

Um gerente pode estar interessado na qualidade de forma geral do que em características específicas da qualidade e, por isso, ele atribuirá métricas, refletindo os requisitos comerciais, para as características individuais.

3. NORMA ISO/IEC 9126

3.1 OBJETIVOS DA NORMA

Considerando que o foco deste trabalho é a avaliação de qualidade de produtos de softwares será descrita em detalhes a norma ISO/IEC 9126 que define seis grandes características que descrevem a qualidade de software. Estas características oferecem uma base para posterior refinamento e descrição de qualidade de software. As diretrizes descrevem o uso de características de qualidade para a avaliação da qualidade de software.

A definição das características e o modelo do processo de avaliação de qualidade desta norma, são aplicáveis na especificação dos requisitos de produtos de software na avaliação de sua qualidade ao longo de seu ciclo de vida.

Suas características são aplicáveis a qualquer tipo de software, incluindo dados e programas de computador contidos em *firmware*.

Segundo [POF1995], a norma ISO/IEC 9126 é um conjunto de padrões dirigida àqueles que estão envolvidos com aquisição, desenvolvimento, uso, suporte, manutenção e auditoria de software.

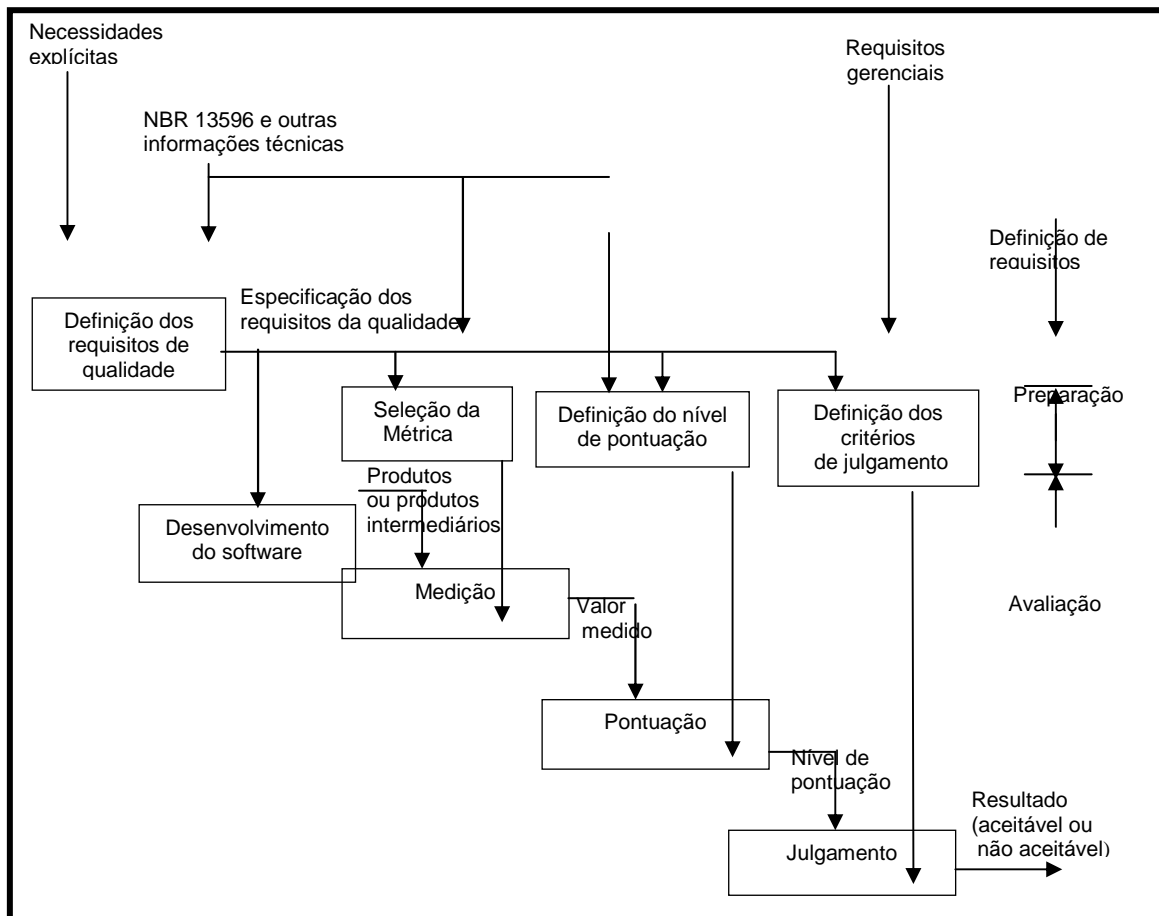
As organizações podem usar estes padrões em grande número de aplicações:

- a) na possibilidade de determinação, pela organização, na capacidade de determinado software suportar um padrão internacional reconhecido;
- b) para auxiliar a organização a melhorar seu próprio processo de desenvolvimento e manutenção de software;
- a) como autojulgamento, para auxiliar uma organização a determinar sua capacidade de implementar um novo projeto de software.

3.2 MÉTODO DE AVALIAÇÃO

Segundo a norma ISO/IEC 9126 e [ELI1999], o processo de avaliação de qualidade é constituído dos seguintes estágios: Definição de Requisitos de Qualidade da Avaliação, Especificação e Preparação da Avaliação, Execução da Avaliação. Este processo pode ser aplicado em cada fase apropriada do ciclo de vida de cada componente do produto de software.

O esquema representado na figura 1 apresenta os passos mais significativos no processo de avaliação de um software segundo a norma ISO/IEC 9126.



Fonte: [ISO1996]

3.2.1 DEFINIÇÃO DE REQUISITOS DE QUALIDADE

No estágio inicial são especificados os requisitos em termos de características da qualidade e possíveis subcaracterísticas. Os requisitos expressam às exigências de ambiente pelo produto de software em consideração, e precisam ser definidos antes do desenvolvimento, como:

- definir o objetivo da avaliação;
- identificar o produto a ser avaliado;
- definir os requisitos (funcionais/operacionais) do produto.

3.2.2 ESPECIFICAÇÃO E PREPARAÇÃO DA AVALIAÇÃO

O objetivo do segundo estágio é preparar as bases para a avaliação e compõe-se das seguintes etapas:

a) seleção de atributos:

- os atributos são propriedades que evidenciam a presença de uma determinada característica da qualidade em um produto de software;

b) seleção de métricas da qualidade:

- da forma como as características da qualidade são definidas não se permite sua medição direta. É necessário estabelecer métricas correlacionadas às características do produto de software. Toda particularidade quantificável do software e toda interação quantificável do software com o seu ambiente que esteja relacionada a uma característica podem ser estabelecidas como uma métrica;
- as métricas podem depender do ambiente e das fases do processo de desenvolvimento que são usadas;

c) definição dos níveis de pontuação:

- o resultado, é mapeado em uma escala. Ele não mostra o nível de satisfação. Estas escalas precisam ser divididas em faixas correspondentes aos diversos graus de satisfação dos requisitos. Como a qualidade se refere às necessidades especificadas, não são possíveis níveis de pontuação genéricos. Eles precisam ser definidos para cada avaliação específica;

d) definição dos critérios de julgamento:

- para se julgar a qualidade de um produto de software, os resultados da avaliação das diversas características devem ser sintetizados. O avaliador deve preparar um procedimento para isso. Este procedimento, usualmente, irá abranger outros aspectos, tais como o tempo e custos que contribuem para a avaliação da qualidade de um produto de software em um determinado ambiente;

e) elaboração de um questionário de avaliação:

- o questionário está dividido entre as características e subcaracterísticas a serem avaliadas;

- além das métricas que estão em forma de questão a serem respondidas, contém informações em relação ao preenchimento do questionário.

3.2.3 EXECUÇÃO DA AVALIAÇÃO

Este estágio do Modelo do Processo de Avaliação que realiza as medições do produto é composto por três passos:

a) medição:

- para a medição, as métricas escolhidas são aplicadas ao produto de software. O resultado é dado em valores na escala das métricas;

b) pontuação:

- o nível é determinado para o valor medido;

c) julgamento:

- é o passo final no processo de avaliação, onde um conjunto de níveis é sintetizado. O resultado é uma declaração da qualidade do produto de software. Esta qualidade sintetizada é comparada com outros aspectos, como tempo e custo. Finalmente a decisão gerencial será tomada com base em critérios gerenciais. O resultado é uma decisão gerencial quanto à aceitação ou rejeição, ou quanto à liberação ou não-liberação do produto de software;

Exemplo: valor medido – nível pontuado (excelente) = satisfatório

valor medido – nível pontuado (insuficiente) = insatisfatório

A execução gera os seguintes resultados:

- relatório com a pontuação alcançada pelo produto final;
- relatório por característica e subcaracterística da qualidade.

3.3 QUALIDADE DE PRODUTO DE SOFTWARE

A qualidade de um produto de software é resultante das atividades realizadas no processo de desenvolvimento do mesmo. Avaliar a qualidade de um produto de software é

verificar, através de técnicas e atividades operacionais o quanto os requisitos são atendidos. Tais requisitos, de uma maneira geral, são a expressão das necessidades explícitas em termos quantitativos ou qualitativos, e têm por objetivo definir as características de um software, a fim de permitir o exame de seu atendimento [TSU1997].

As organizações internacionais de normalização ISO/IEC vêm trabalhando juntamente em um modelo que permite avaliar a qualidade dos produtos de software. O processo de avaliação segundo [WEB1997], é definido pelas seguintes normas internacionais:

- a) características da qualidade e métricas:
 - ISO/IEC 9126-1: Características e Subcaracterísticas da Qualidade;
 - ISO/IEC 9126-2: Métricas Externas;
 - ISO/IEC 9126-3: Métricas Internas;

- b) avaliação dos produtos de software:
 - ISO/IEC 14598-1: Visão Geral;
 - ISO/IEC 14598-2: Planejamento e Gerenciamento;
 - ISO/IEC 14598-3: Processo para Equipe de Desenvolvimento;
 - ISO/IEC 14598-4: Processo para Adquirentes a ISO/IEC;
 - ISO/IEC 14598-5: Processo para Avaliadores;
 - ISO/IEC 14598-6: Módulos de Avaliação;

- c) requisitos da qualidade e teste em pacotes de software;

- d) ISO/IEC 12119: Pacotes de Software – requisitos da qualidade e teste.

Este trabalho tratará apenas das características da qualidade da norma ISO/IEC 9126-1.

3.3.1 CARACTERÍSTICAS DA QUALIDADE E MÉTRICAS – ISO/IEC 9126-1

Segundo [WEB1997], esta série de documentos padroniza as definições de características da qualidade, subcaracterísticas e métricas que são recomendadas para serem usadas para a especificação dos requisitos da qualidade, planejamento de lista de checagem para revisão e teste final para o projeto.

De acordo com as normas da ISO/IEC 9126, a avaliação de um produto de software deveria ser:

- a) **repetitiva:** repetindo a avaliação sobre um mesmo produto com a mesma especificação de avaliação, pela mesma equipe de testes, deve produzir o mesmo resultado;
- b) **reproduzível:** repetindo a avaliação de um mesmo produto com a mesma especificação por diferentes equipes de teste deve dar o mesmo resultado;
- c) **imparcial:** a avaliação deve ser livre de tendências injustas.

O critério proposto pela ISO/IEC 9126 consta de seis grandes características principais as quais divididas em um conjunto de subcaracterísticas, a seguir descreve-se todas as características com suas respectivas divisões e conceitos [ISO1996].

3.3.1.1 FUNCIONALIDADE

Conjunto de atributos de software que influenciam a existência de um conjunto de funções que satisfaçam necessidades implícitas, e suas propriedades específicas. As funções são aquelas que satisfazem estados ou necessidades implicadas.

Subcaracterísticas da funcionalidade:

- a) atributos de Adequação: atributos de software que influenciam na presença de adequação de um conjunto de funções para tarefas específicas e objetivos do uso;
- c) atributos de Acurácia: atributos do software que evidenciam a presença de resultados ou efeitos corretos ou conformidade acordados;
- c) atributos de Interoperabilidade: atributos de software que influenciam na habilidade de interagir com um ou mais sistemas específicos;

- d) atributos de Conformidade: Atributos do software que influenciam na aderência e padrões relativos a convenções ou regulamentações legais e prescrições similares;
- e) atributos de Segurança: Atributos de software que influenciam na habilidade de prevenir acessos não intencionados e resistir ataques intencionados para se ter acesso não autorizado a informação confidencial, ou fazer modificações não autorizadas em informações ou em programa;

3.3.1.2 CONFIABILIDADE

Conjunto de atributos de software que influenciam na capacidade do software em manter seu nível de desempenho sob condições específicas, para um período de tempo específico.

Subcaracterísticas da confiabilidade:

- a) atributos de Maturidade: atributos de software que influenciam na frequência de erros devido à falhas no software;
- b) atributos de Tolerância à Falhas: atributos de software que influenciam na habilidade de um nível específico de desempenho em casos de falhas do software ou por violação de sua interface específica;
- c) atributos de Recuperabilidade: atributos de software que influenciam sua capacidade de restabelecer seu nível de desempenho e recuperar os dados diretamente afetados no caso de ocorrer uma falha e no tempo e esforço necessários.

3.3.1.3 USABILIDADE

Conjunto de atributos de software que influenciam na capacidade do software contido no sistema em ser fácil de usar e satisfazer o usuário, sob condições específicas.

Subcaracterística de usabilidade:

- a) atributos de Inteligibilidade: atributos de software que influenciam na capacidade do usuário entender se o software é adequado, e como ele pode ser usado para tarefas e condições de uso particulares;
- b) atributos de Apreensibilidade: atributos de software que influenciam a facilidade com a qual o usuário pode aprender suas aplicações;
- c) atributos de Operacionalidade: atributos de software que influenciam no esforço necessário para o usuário poder operar e manter o controle da operação.

3.3.1.4 EFICIÊNCIA

Conjunto de atributo de software que influenciam na performance requerida no software e a qualidade dos recursos usados, sob condições específicas.

Subcaracterísticas da eficiência:

- a) atributos de Comportamento em Relação ao Tempo: atributos do software que influenciam no tempo de resposta e processamento e desempenho na execução de sua funções;
- b) Atributos de Comportamento em Relação aos Recursos: atributos de software que influenciam na quantidade de recursos usados e a duração de tal uso na execução de sua funções.

3.3.1.5 MANUTENIBILIDADE

Conjunto de atributos de software que influenciam nos recursos necessários para fazer modificações específicas.

Subcaracterísticas da manutenibilidade:

- a) atributos de Analisabilidade: atributos de software que influenciam na necessidade de recursos necessários para diagnóstico de deficiências ou causas de falhas, ou para identificação de partes a serem modificadas;
- b) atributos de Modificabilidade: atributos de software que influenciam na necessidade de recursos para implementar as modificações específicas;
- c) atributos de Estabilidade: atributos de software que influenciam no risco de efeitos inesperados das modificações;
- d) atributos de Testabilidade: atributos de software que influenciam na necessidade de recursos necessários para validação do software modificado;

3.3.1.6 PORTABILIDADE

Conjunto de atributos de software que influenciam na habilidade do software ser transferido de um ambiente para outro.

Subcaracterísticas de portabilidade:

- a) atributos de Adaptabilidade: atributos de software que influenciam na capacidade e esforço necessário para sua adaptação em ambientes diferentes especificados, sem aplicar outros meios ou ações para atingir propósito para o software;
- b) atributos de Instalação: atributos de software que influenciam o esforço necessário para instalar o software no ambiente especificado;

- c) atributos de Conformidade: atributos de software que fazem o software manter padrões ou convenções relativas comuns;

- d) atributos da Capacidade de Substituição: atributos de software que proporcionam a oportunidade e influenciam no esforço requerido para usá-lo em lugar de outro software específico no ambiente de tal software.

4 PROPOSTA DE MENSURAÇÃO DA AVALIAÇÃO DA QUALIDADE DE SOFTWARE - ISO/IEC 9126

O processo de avaliação da qualidade de produtos de software proposto tem como base os requisitos de qualidade propostos pela Norma ISO/IEC 9126. Os requisitos definidos no capítulo 3 não podem ser aplicados diretamente na avaliação, havendo a necessidade de um detalhamento maior das características aplicadas para melhor compreensão do que está sendo avaliado.

Assim, cada requisito que foi detalhado engloba uma hierarquia de características, obedecendo sempre o propósito do requisito anterior.

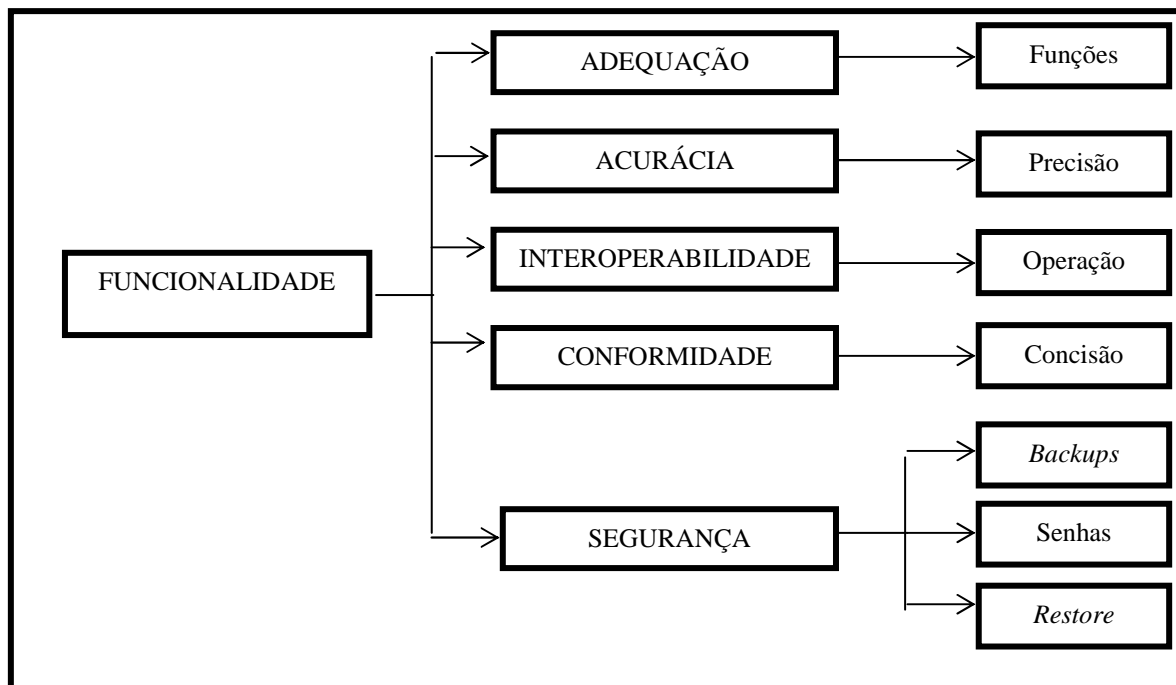
O processo de montagem do modelo de avaliação foi realizado com o objetivo de levantar requisitos da qualidade que deveriam estar presentes em um software de qualquer área.

A seguir apresenta-se a proposta de detalhamento da norma, com a elaboração de um diagrama hierárquico contendo as seis características de primeiro nível subdivididas para melhor compreensão. Além disso, são apresentados os atributos relacionados a cada subcaracterística. Os diagramas foram baseados no modelo de avaliação descrito em [GUE1995]. No Anexo A encontra-se o detalhamento de cada diagrama.

4.1 FUNCIONALIDADE

Refere-se à existência de um conjunto de funções que satisfaçam às necessidades explícitas e implícitas, e suas propriedades específicas. Segue o detalhamento conforme a figura 2.

FIGURA 2 - DIAGRAMA DA ESTRUTURA DA CARACTERÍSTICA FUNCIONALIDADE



a) atributos de Adequação

→ avalia a presença e adequação de um conjunto de funções para tarefas específicas.

- **funções:**

Requisitos mínimos que um software deve ter para executar todas as tarefas para as quais foi desenvolvido.

b) atributo de Acurácia

→ avalia a precisão de resultados e efeitos corretos das funções necessárias para um software de qualquer área.

- **precisão:**

Requisitos mínimos que um software precisa ter para fornecer resultados certos. Refere-se a precisão os valores manipulados pelo sistema. Exemplo: o grau necessário de precisão nos cálculos. Resultado ou efeito corretos das funções.

c) atributos de Interoperabilidade

→ avalia os atributos do software na capacidade de interagir com outros softwares.

- **operação:**

Requisitos mínimos necessários que um software precisa ter para poder integrar com outros programas, importar e exportar dados, capacidade de rodar em outros ambientes e capacidade de operação em rede.

d) atributos de Conformidade

→ avalia a conformidade do software em relação às leis vigentes.

- **concisão:**

Requisitos mínimos que um software precisa ter em relação a padrões funcionais, a legislação e regulamentação legal.

e) atributos de Segurança

→ avalia a habilidade de qualquer software com relação a proteção de dados, como prevenir contra acesso não autorizado a programas e dados.

- **backups:**

Processo de geração e tratamento de backups.

- **senhas:**

Capacidade do software evitar acesso a programas e dados de usuários não autorizados.

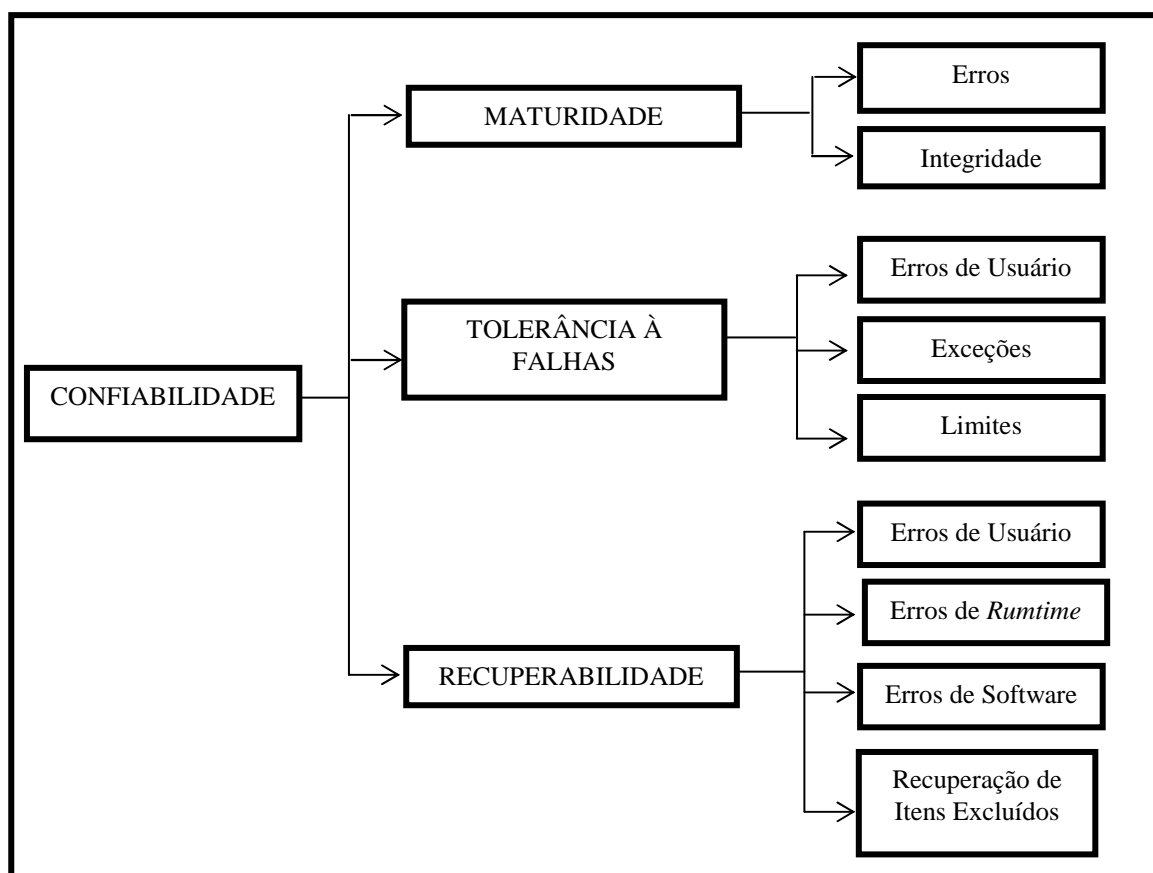
- **restore:**

Capacidade do software manter a restauração de dados e informações.

4.2 CONFIABILIDADE

É uma série de atributos que suporta a capacidade do software manter seu nível de desempenho, performance sob condições estabelecidas, ou quando da ocorrência de falhas e a frequência da ocorrência de falhas. Segue o detalhamento conforme a figura 3.

FIGURA 3 – DIAGRAMA DA ESTRUTURA DA CARACTERÍSTICA CONFIABILIDADE



a) atributos da Maturidade

→ avalia a capacidade do software em manter seu nível de performance, a frequência de erros devido à falhas no software. Aborda a precisão dos dados, a ocorrência de erros durante a utilização do software e controle de integridade.

- **erros:**

Probabilidade de ocorrer erros durante a execução de funções específicas, atividades de configuração do sistema e erros de entrada de dados.

- **integridade:**

Refere-se à capacidade do software manter a integridade dos dados após a intervenção de usuários, queda de energia e erros de execução.

b) atributos de Tolerância à Falhas

→ avalia a capacidade de qualquer software em manter seu nível de performance quando da ocorrência de falhas no software. Avalia a capacidade do software controlar as ações do usuário, tratando de situações anormais e o desempenho do sistema com relação ao volume de dados.

- **erros de usuário:**

Probabilidade de ocorrer erros de preenchimento de campos, exclusão de dados existentes, configuração de software e impressora.

- **exceções:**

Capacidade do software em tratar situações anormais que possam ocorrer durante execução do sistema.

- **limites:**

Capacidade do software manter seu desempenho, usando grande volume de dados e transações.

c) atributos de Recuperabilidade

→ avalia a capacidade do software de reestabelecer seu nível de desempenho e recuperar os dados diretamente afetados em caso da ocorrência de falhas, no prazo e esforço necessário para isto.

- **erros do usuário:**

Capacidade do software em permitir a correção de erros cometidos pelo usuário.

- **erros de *Runtime*:**

Capacidade do software em manter sua operação na ocorrência de falhas durante a execução.

- **erros de software:**

Capacidade do software recuperar seus dados no caso de falha no software.

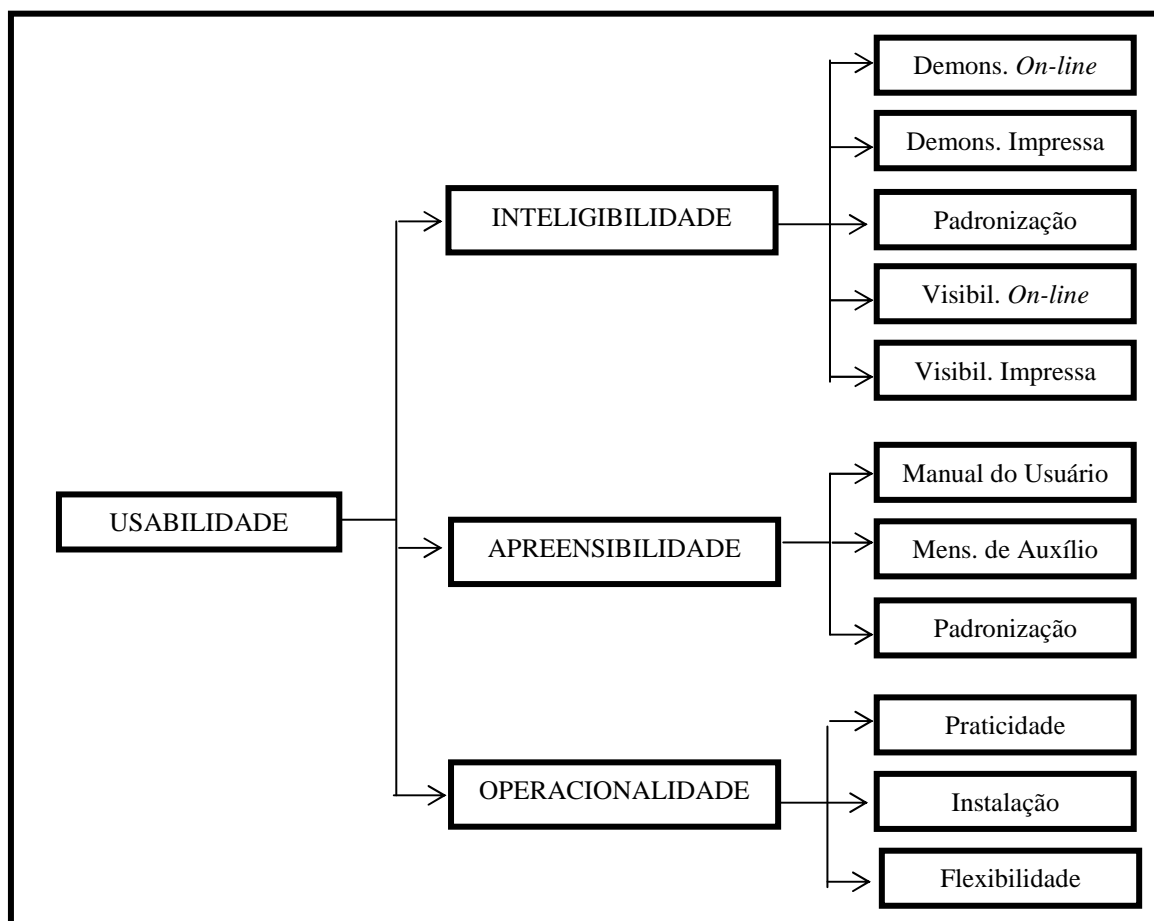
- **recuperação de itens excluídos:**

Capacidade do sistema recuperar dados excluídos incorretamente.

4.3 USABILIDADE

Avalia o esforço necessário para se utilizar o software, operar, aprender, bem como para o julgamento individual desse uso, para um conjunto de usuários implícitos ou explícitos. Segue o detalhamento conforme a figura 4.

FIGURA 4 – DIAGRAMA DA ESTRUTURA DA CARACTERÍSTICA USABILIDADE



a) atributos de Inteligibilidade

→ avalia o esforço do usuário para conhecer os conceitos lógicos do software e sua aplicabilidade.

- **demonstração on-line:**

Autodemonstração e a facilidade de acessar estes recursos de forma *on-line* durante a execução do sistema.

- **demonstração impressa:**

Existência de manual com demonstração de como utilizar o software.

- **padronização:**

Padronização dos elementos de interface em termos de formato, localização de informações na tela, permitindo maior velocidade na realização das tarefas.

- **visibilidade *on-line*:**

Capacidade do software fornecer ao usuário informações sobre a utilização do software, durante a sua utilização.

- **Visibilidade impressa:**

Capacidade do software possuir a documentação impressa sobre o embasamento da teoria em relação a sua utilização.

b) atributos de Apreensibilidade

→ avalia o esforço requerido ao usuário para operar o software.

- **manual do usuário:**

Características do software em meio impresso, facilitando o entendimento e a localização das informações no manual e a utilização das informações nele contidas.

- **mensagem de auxílio:**

Facilidade de localizar as mensagens de auxílio com clareza.

- **padronização:**

Facilidade de uso do sistema pelo usuário devido a padronização das ações e interfaces e menus.

c) atributos de Operacionalidade

→ avalia o esforço do usuário para a operação e controle de operação das funções do software.

- **praticidade/navegação:**

Facilidade de movimentação dentro do sistema, no acesso aos menus e na execução das funções.

- **instalação:**

Capacidade do software realizar a instalação do sistema e interagir com o usuário.

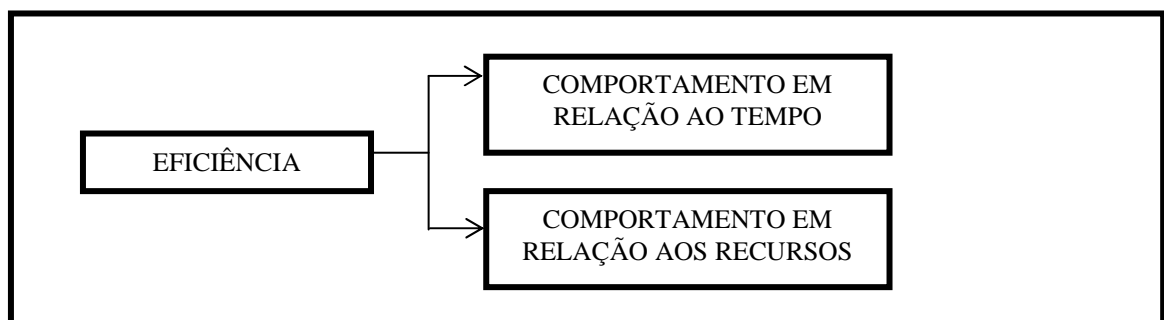
- **flexibilidade:**

Capacidade do software permitir ao usuário de realizar modificações no sistema.

4.4 EFICIÊNCIA

Avalia a relação entre o nível de desempenho do software e a quantidade de recursos usados, sob condições estabelecidas. Segue o detalhamento conforme a figura 5.

FIGURA 5 – DIAGRAMA DA ESTRUTURA DA CARACTERÍSTICA EFICIÊNCIA



a) atributos do Comportamento em Relação ao Tempo

→ avalia o tempo de processamento e respostas e a taxa de tempo no desempenho de sua função.

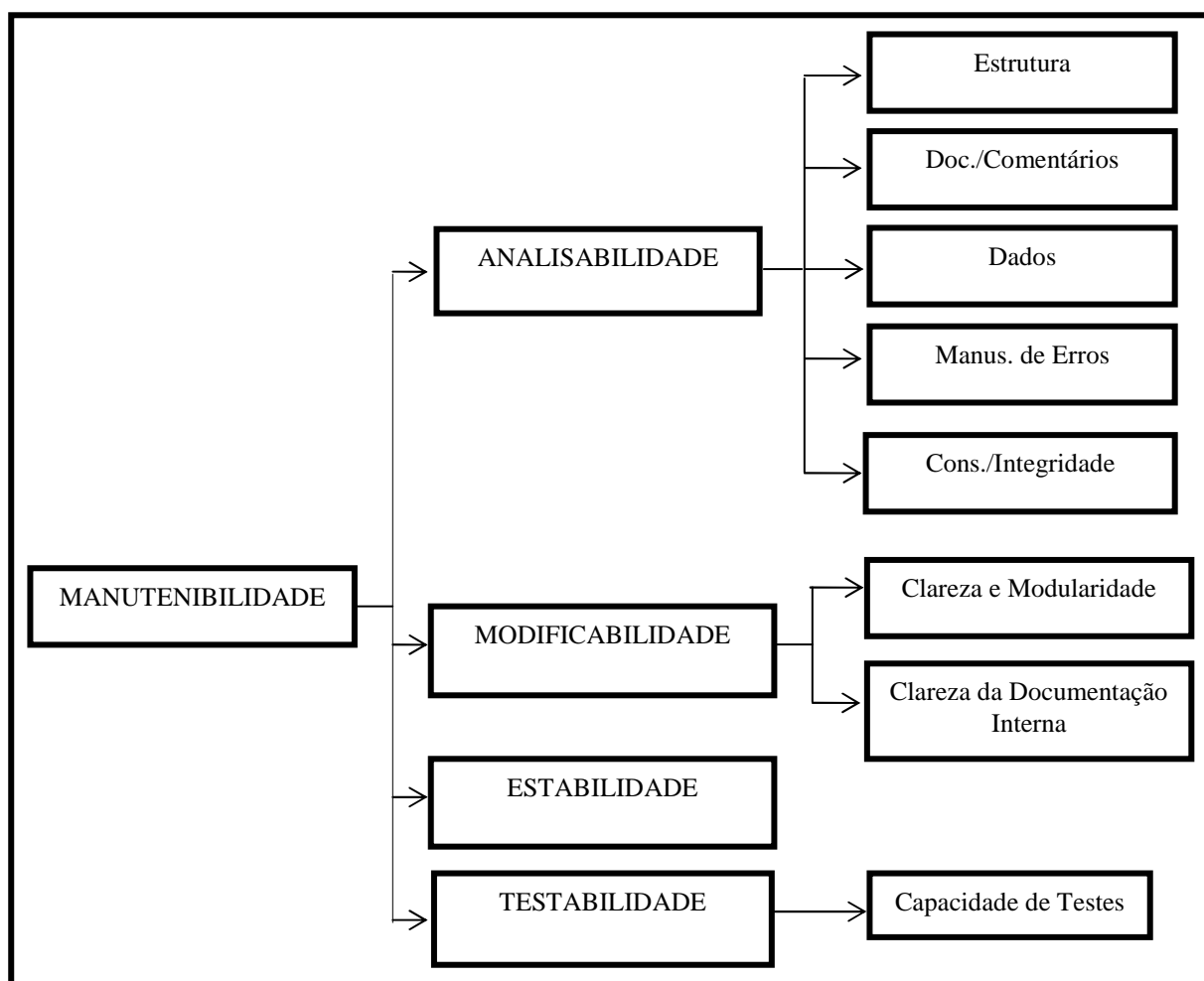
b) atributos do Comportamento em Relação aos Recursos

→ avalia a quantidade de recursos utilizados para atingir o nível de desempenho de uma função desejada pelo software.

4.5 MANUTENIBILIDADE

Avalia o esforço necessário para fazer modificações específicas no software. Segue o detalhamento conforme a figura 6.

FIGURA 6 – DIAGRAMA DA ESTRUTURA DA CARACTERÍSTICA MANUTENIBILIDADE



a) atributos de Analisabilidade

→ avalia o esforço necessário para diagnosticar deficiências ou causas de falhas, ou identificação de partes a serem modificadas.

- **estrutura:**

Forma de como o sistema foi desenvolvido, a padronização utilizada para a definição de identificadores, formato de módulos do sistema e convenções de codificação.

- **documentação/comentários:**

Existência de comentários descrevendo cada variável, unidade e processo existente no sistema. O código do sistema como um todo.

- **dados:**

Definição correta de variáveis e tabelas.

- **manuseio de erros:**

Capacidade do software tratar os erros ocorridos durante o desenvolvimento, como testes de consistência em nível de código fonte para evitar a ocorrência do mesmo.

- **consistência/integridade:**

Relação entre o programa e o projeto, a importância de cada programa no funcionamento do sistema. Verifica se o sistema foi elaborado consistentemente.

b) atributos de Modificabilidade

→avalia esforço necessário para fazer modificações, remoção de erros, ou para modificações de ambientes.

- **clareza e modularidade do software:**

Capacidade do software se adaptar as mudanças possíveis na legislação e no próprio sistema.

- **clareza da documentação interna do software:**

Existência de uma documentação bem elaborada e clara do software.

c) atributos de Estabilidade

→ avalia os riscos de efeitos inesperados devido a modificações. Avalia os erros que podem surgir com as modificações.

- **capacidade de testes:**

Capacidade do software suportar e validar as modificações no sistema.

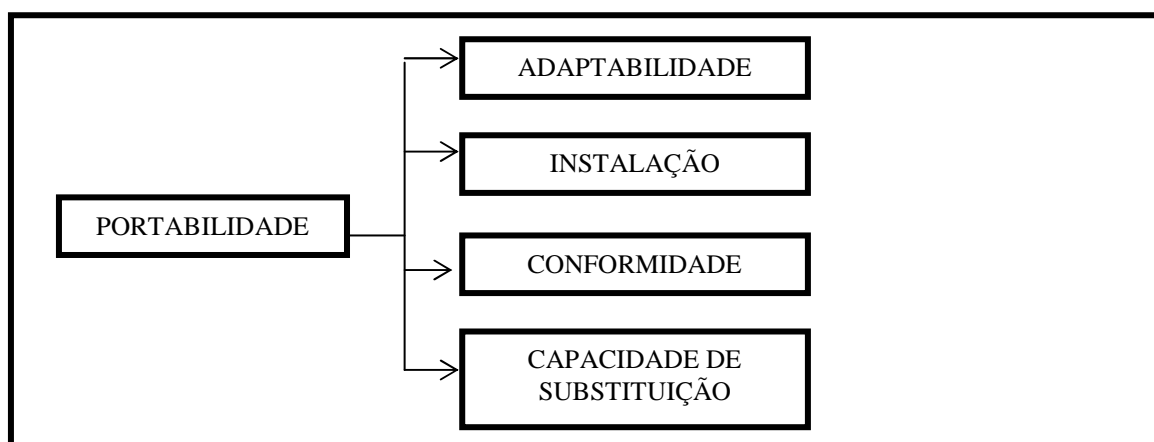
d) atributos de Testabilidade

→ avalia o esforço necessário para validação do software modificado.

4.6 PORTABILIDADE

Avalia os atributos que suportam a habilidade do software ser transferido de um ambiente para outro qualquer, ser manipulado e operado de maneira fácil e adequado em configuração de equipamentos diferentes do original. Segue o detalhamento conforme a figura 7.

FIGURA 7 – DIAGRAMA DA ESTRUTURA DA CARACTERÍSTICA PORTABILIDADE



a) atributos de Adaptabilidade

→ avalia os atributos do sistema que suportam a habilidade de adaptação do software para diferentes ambientes sem aplicar outras ações ou meios do que provem os propósitos do software considerado.

b) atributos de Instalação

→ avalia o esforço necessário para instalar o software em ambiente específico. Inclui os recursos de auxílio fornecidos pelo software.

c) atributos de Conformidade

→ avalia os atributos do software em manter padrões ou convenções relativas a portabilidade.

d) atributos de Capacidade de Substituição

→ avalia a habilidade do software ser usado no lugar de outro software especificado no ambiente deste software.

5. DESENVOLVIMENTO DO PROTÓTIPO

5.1 INTRODUÇÃO

Através do detalhamento da Norma ISO/IEC 9126, tornou-se possível o desenvolvimento do protótipo para a avaliação da qualidade de um software de qualquer área, conforme o objetivo deste trabalho.

Para a implementação do protótipo, foram analisadas as principais características que um software de qualquer área possa possuir para atender às necessidades de seus usuários. A partir da Norma ISO/IEC 9126, foi elaborado um questionário com uma série de perguntas que é a base para toda a avaliação. Os requisitos de qualidade utilizados para a elaboração do questionário foram obtidos em avaliações de softwares específicos [SAN1998], [PISK1996], [ELI1999] e observando os principais requisitos que um software precisa ter para satisfazer o usuário.

5.2 AMBIENTE DE DESENVOLVIMENTO

Para a especificação do protótipo foi utilizada a ferramenta CASE System Architect 2001 e o protótipo foi implementado utilizando-se o ambiente Delphi 5.0.

5.3 SYSTEM ARCHITECT

Segundo [CHO1999], o System Architect 2001 (SA/2001) é uma ferramenta CASE voltada para a análise/projeto/correção podendo ser classificada como uma ferramenta de Documentação de Sistemas e Engenharia de Informação, atuando em Análise, Projeto Preliminar, Modelagem de Informações, Projeto Detalhado, Acompanhamento de Projeto, Documentação de dados existentes, etc., o que permite muitas empresas alcançar a produtividade, qualidade e integração.

O System Architect trabalha em várias plataformas de comunicação a qual o Windows se conecta, permitindo o uso em grupos de trabalho através do compartilhamento das áreas de trabalho. Ele permite a utilização de várias técnicas durante o ciclo de desenvolvimento de um projeto, como Diagrama de Fluxo de dados, Entidade e Relacionamento, Diagrama de Estado de Transição, etc.

Além das técnicas disponíveis para o desenvolvimento, o System Architect, permite documentar informações relativas a modelagem de dados - Banco de Dados (SyBase, ORACLE, DB2 e INFORMIX,) e Front_Ends (Prototipação, PowerBuilder e SQL). Ele gera *SCRIPTS* para base SQL através das informações no dicionários de dados do SA/2001.

Em relação a portabilidade, o SA/2001, possibilita a comunicação com vários softwares conhecidos, como Delphi, Power Builder, Visual Basic e Word .

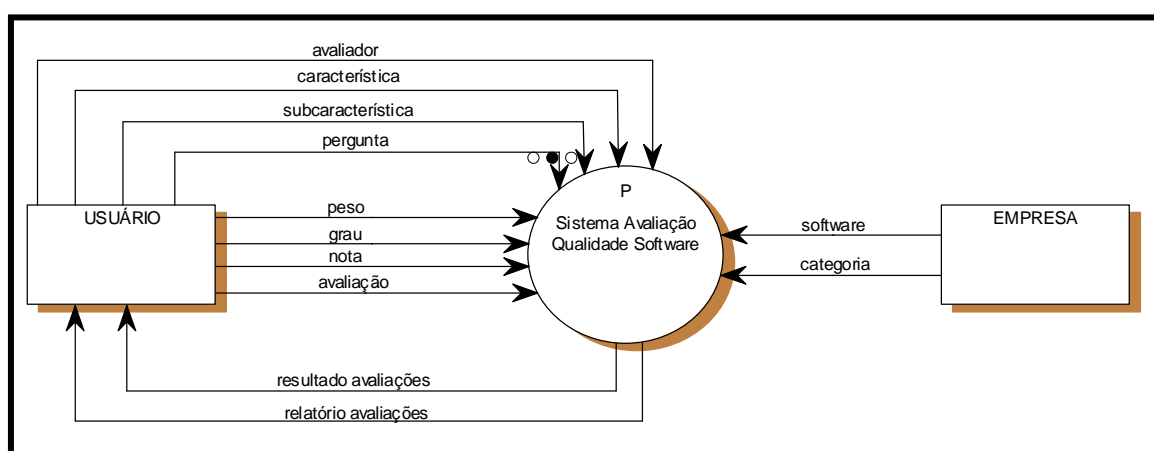
Através das facilidade de customização, a ferramenta atende à várias fases do ciclo de desenvolvimento. A sua flexibilidade de configuração o torna extremamente abrangente, modelando informações às necessidades da metodologia utilizada.

5.4 DIAGRAMA DE FLUXO DE DADOS

O Diagrama de Fluxo de Dados identifica as entidade externas com as quais o sistema se comunica e declara o objetivo principal do sistema, na forma de um só processo. Ele mostra os componentes ativos do sistema e as interfaces de dados entre eles. O DFD escreve detalhadamente todos os elementos da modelagem.

Na figura 8 é apresentado o diagrama de contexto e na figura 9 o diagrama de fluxo de dados nível 0.

FIGURA 8 – DIAGRAMA DE CONTEXTO



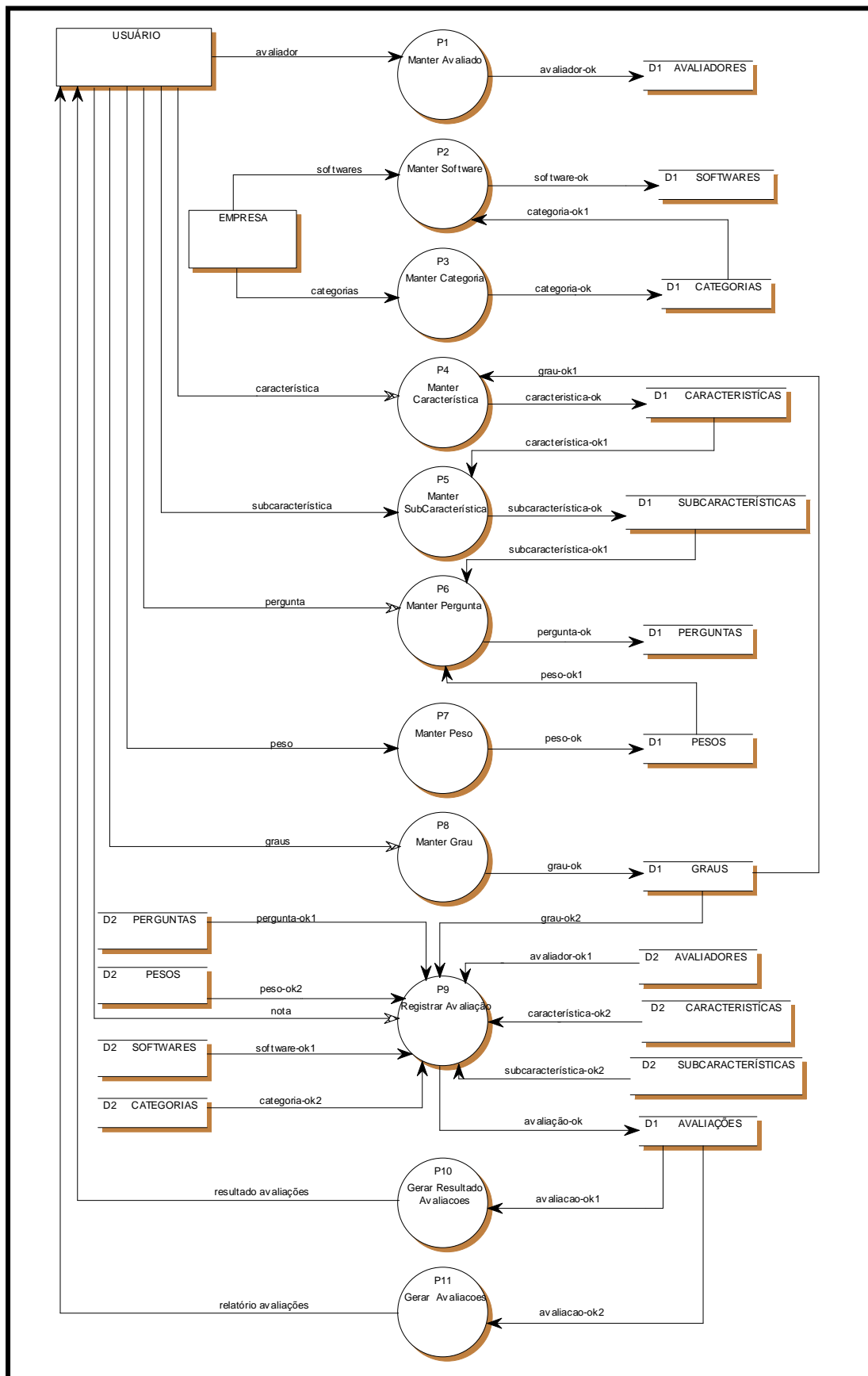
No diagrama de contexto pode-se perceber que o sistema de avaliação recebe os dados cadastrados para realizar a avaliação e retorna o resultados para o avaliador.

O processo 9 do diagrama de Fluxo de Dados Nível 0 funciona da seguinte forma:

- a) o usuário/avaliador entra com os dados do avaliador, os dados do software, a categoria do software, com as perguntas do questionário de avaliação, define os pesos, o grau de importância de cada característica, configura a avaliação, isto é, seleciona as características e subcaracterísticas que serão avaliadas, inicia o processo de avaliação respondendo o questionário, dando uma nota para cada pergunta;
- b) o sistema recebe as informações, processa e registra a avaliação .

No processo 10 do diagrama de Fluxo de Dados Nível 0, o sistema retorna o resultado da avaliação para o avaliador, e no processo 11 o sistema emite os relatórios. Os relatórios implementados são os seguintes:

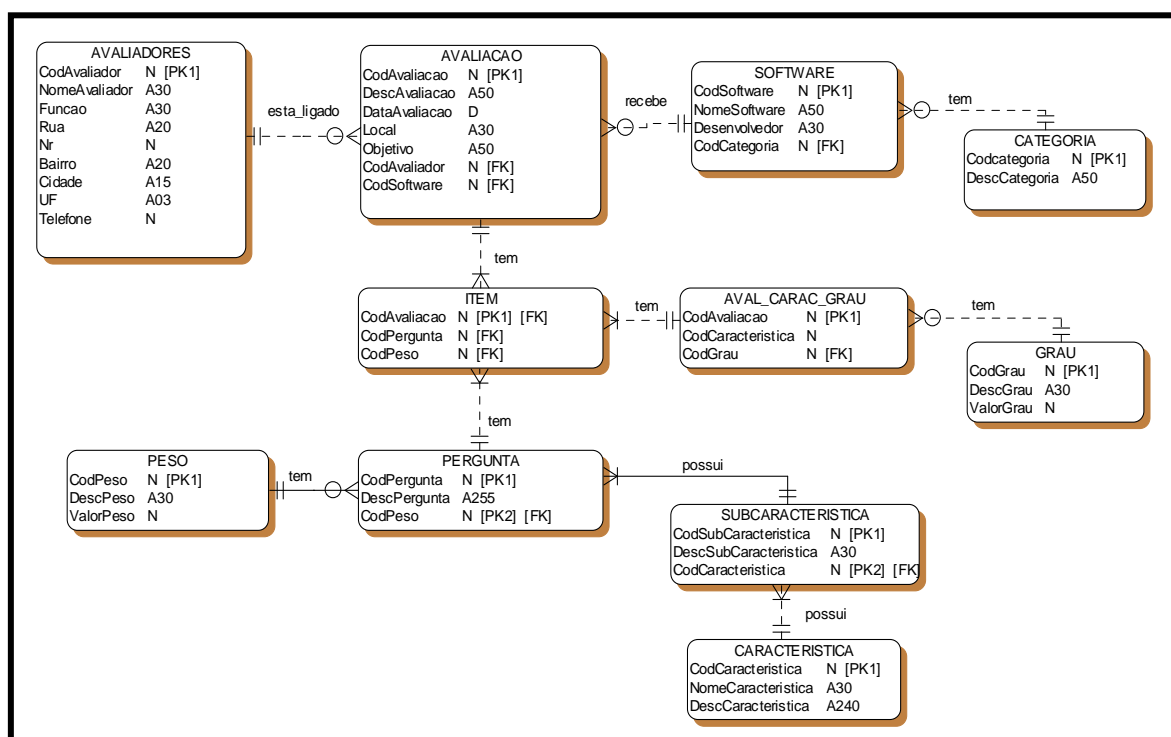
- a) relatório de todas as avaliações realizadas;
- b) relatório com o resultado individual de cada avaliação.



5.5 MODELO ENTIDADE E RELACIONAMENTO

O modelo entidade relacionamento (MER), enfatiza os principais objetivos ou entidades do sistema. Detalha as associações existentes entre as Entidades de Dados e utiliza componentes semânticos próprios. O Modelo entidade Relacionamento é apresentado na figura 10.

FIGURA 10 – MODELO ENTIDADE E RELACIONAMENTO



O Modelo Entidade e Relacionamento é composto por onze entidades associadas entre si, conforme a implementação do protótipo. A entidade AVALIAÇÃO representa todos os atributos referente a uma avaliação realizada e está associada a entidade AVALIADOR através do CodAvaliador, a entidade SOFTWARE através do CodSoftware, e a entidade ITEM através do CodAvaliação. A entidade SOFTWARE está associada a entidade CATEGORIA através do CodCategoria. A entidade PERGUNTA está associada as entidades PESO (CodPeso), SUBCARACTERÍSTICA (CodSubcaracterística) e ITEM (CodPergunta). A entidade SUBCARACTERÍSTICA está associada a entidade CARACTERÍSTICA através do CodCaracterística. Cada característica pode ter várias subcaracterísticas e cada subcaracterística pode ter várias

perguntas associadas. A entidade AVAL_CARACT_GRAU está associada a entidade ITEM através do CodAvaliação, e a entidade Grau através do CodGrau.

5.6 DICIONÁRIO DE DADOS

O Dicionário de Dados contém a definição de todos os dados mencionados no MER, as entidades e seus atributos, incluindo detalhes do formato físico, como: tipo, tamanho, chave e descrição do atributo. Gerou-se um relatório na ferramenta CASE System Architect, utilizada para o desenvolvimento da modelagem, através do *SCRIPTS*. A seguir estão descritas as tabelas de cada entidade com seus respectivos atributos.

ENTIDADE: AVAL_CARAC_GRAU

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodAvaliacao	N	03	*
CodCaracteristica	N	03	
CodGrau	N	03	

ENTIDADE: AVALIACAO

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodAvaliacao	N	03	*
DescAvaliacao	A	30	
DataAvaliacao	D	03	
Local	A	50	
Objetivo	A	50	
CodAvaliador	N	03	

CodSoftware	N	03	
--------------------	----------	-----------	--

ENTIDADE: AVALIADORES

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodAvaliador	N	03	*
NomeAvaliador	A	30	
Funcao	A	30	
Rua	A	20	
Nr	N	03	
Bairro	A	20	
Cidade	A	15	
UF	A	3	
Telefone	N	03	

ENTIDADE: CARACTERÍSTICA

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodCaracteristica	N	03	*
NomeCaracteristica	A	30	
DescCaracteristica	A2	40	

ENTIDADE: CATEGORIA

ATRIBUTO	TIPO	TAMANHO	CHAVE
Codcategoria	N	03	*
DescCategoria	A	50	

ENTIDADE: GRAU

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodGrau	N	03	*
DescGrau	A	30	
ValorGrau	N	03	

ENTIDADE: ITEM

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodAvaliacao	N	03	*
CodPergunta	N	03	

ENTIDADE: PERGUNTA

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodPergunta	N	03	*
CodPeso	N	03	*
DescPergunta	A	255	
CodSubCaracteristica	N	03	

ENTIDADE: PESO

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodPeso	N	03	*
DescPeso	A	30	
ValorPeso	N	03	

ENTIDADE: SOFTWARE

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodSoftware	N	03	*
NomeSoftware	A	50	
Desenvolvedor	A	30	
CodCategoria	N	03	

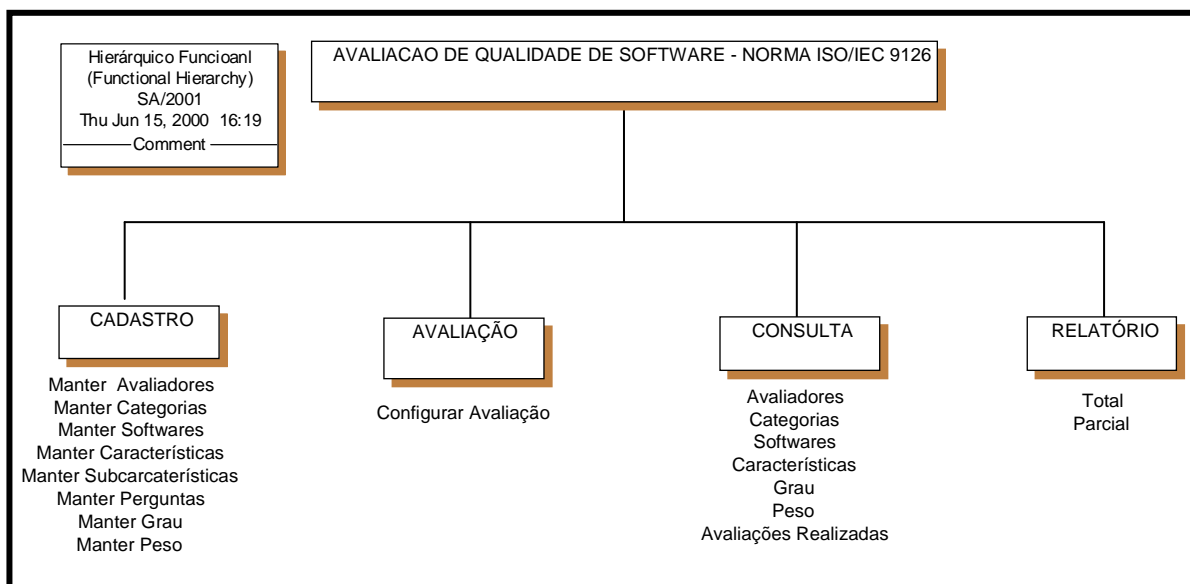
ENTIDADE: SUBCARACTERISTICA

ATRIBUTO	TIPO	TAMANHO	CHAVE
CodSubCaracteristica	N	03	*
CodCaracteristica	N	03	*
DescSubCaracteristica	A	30	

5.7 DIAGRAMA HIERÁQUICO FUNCIONAL

O Diagrama Hierárquico Funcional do protótipo é apresentado na figura 11. Ele consiste de um menu de “Cadastro” onde o usuário terá os principais cadastramentos do sistema, tais como: Cadastro de Avaliadores, Cadastro de Categorias, Cadastro de Software, Cadastro de Características e Subcaraterísitcas, Cadastro de Perguntas, Cadastro de Graus e Cadastro de Pesos. No menu “Avaliação” ocorre todo o processo de avaliação do software. No menu “Consulta” o avaliador terá todas as opções de consultas implementadas no protótipo e no menu “Relatórios” encontram-se todos os relatórios com os resultados obtidos na avaliação.

FIGURA 11 - DIAGRAMA HIERÁRQUICO FUNCIONAL



5.8 DESCRIÇÃO DO PROTÓTIPO

O protótipo desenvolvido tem por objetivo avaliar a qualidade de um produto de software em qualquer área, dando suporte as empresas na melhoria da qualidade de seus produtos, visando a satisfação do cliente e o melhoramento contínuo. Segundo a Norma ISO/IEC 926 ainda existem poucas métricas de aceitação geral, o que torna possível estabelecer modelos próprios de processo de avaliação e métodos para a criação e validação de métricas relacionadas com as características dessa norma para atender as diversas áreas de aplicação.

O protótipo contém o cadastro da estrutura de atributos da qualidade conforme a montagem do modelo. Cada atributo será avaliado de acordo com o questionário elaborado, ao qual o avaliador deverá responder escolhendo uma dentre as opções de resposta: Sim, Parcialmente e Não, analisando o grau de conformidade do software em relação ao atributo a ser avaliado.

Cada opção está associada a um peso já definido, mas podendo ser alterado caso haja interesse do avaliador. Da mesma forma, as características e subcaracterísticas podem ser alteradas, ou mesmo, inserir novas características, bem como as perguntas do

questionário de avaliação, adequando a avaliação de acordo com o tipo de software a ser avaliado.

As perguntas que foram implementadas como exemplo no protótipo encontram-se no anexo A.

Através das notas das perguntas referente a cada subcaracterística é calculado o nível de qualidade de cada característica, se a nota dada for sim, terá peso 2, a nota parcialmente terá peso 1 e a nota não terá peso 0.

A fórmula para o cálculo do nível de pontuação ou a qualidade da característica é a seguinte:

$$NQC = \frac{\Sigma N}{\Sigma P}$$

Soma-se as notas das perguntas e divide-se pela soma dos pesos atribuídos para cada pergunta, este peso atribuído tem o valor máximo, considerando que cada pergunta poderá ter nota máxima também, por exemplo, resposta sim. Simplificando, conta-se o número de perguntas e multiplica-se pelo peso máximo (2), para obter o somatório dos pesos.

Exemplo:

$$NQC \text{ (Nível de Qualidade da Característica)} = \frac{\text{Soma Notas}}{\text{Soma Pesos}}$$

Característica: Funcionalidade

Subcaracterística: Adequação

Perguntas:

1. O software possui todas as funções mencionadas no produto?

Sim Parcialmente Não

2. O software dispõe de todas as funções necessárias para a sua execução?

Sim Parcialmente Não

Subcaracterística: Acurácia**Perguntas:**

1. O software é preciso na execução das funções?
 Sim Parcialmente Não

Subcaracterística: Interoperabilidade**Perguntas:**

1. O software tem capacidade para processamento multiusuário?
 Sim Parcialmente Não
2. O software tem capacidade para operação com redes?
 Sim Parcialmente Não

Resultado da avaliação da característica funcionalidade:

Soma das Notas = 9 (o valor 9 é o somatório das notas das perguntas)

Soma dos Pesos = 10 (o valor 10 é a soma máxima dos pesos atribuídos para cada pergunta)

Cálculo: $9 / 10 = 0,9$

Resultado = 0,9 => 90%

O resultado final do nível de qualidade do software é calculado aplicando-se um grau de importância para cada característica da qualidade, podendo ser alterado caso haja interesse do avaliador.

O protótipo tem como padrão de grau de importância a faixa aplicada na tabela de avaliação de produto de software – parte 4: via do comprador [ISO1995], que adota o seguinte padrão:

Funcionalidade = Alto
Confiabilidade = Baixo
Usabilidade = Alto

Eficiência = Médio

Manutenibilidade = Baixo

Portabilidade = Baixo

O valor numérico com o grau de importância está definido da seguinte forma:

Baixo = 1

Médio = 2

Alto = 3

O cálculo da qualidade final do produto, será da seguinte forma:

$$QFP = \frac{\Sigma (NQC * GIC)}{\Sigma GIC}$$

Multiplica-se as média da característica calculada no exemplo acima (NQC), pelo grau de importância atribuído para cada característica, conforme a definição da ISO/IEC 9126, descrito acima, e divide-se pelo somatório do grau de importância de todas as características.

Exemplo:

$$QFP \text{ (Qualidade Final do Produto) } = \frac{\Sigma (\text{Nível de Qualidade Funcionalidade} * \text{Grau de Importância Funcionalidade}) + (\text{Nível de Qualidade Confiabilidade} * \text{Grau de Importância Confiabilidade}) \dots}{\Sigma (\text{Grau de Importância Funcionalidade} + \text{Grau de Importância Confiabilidade})..}$$

Característica: Funcionalidade

Nível de Qualidade de Funcionalidade = 0,9

Grau de Importância = 3

Cálculo: $(0,9 * 3 = 0,27 / 3 = 0,9)$

Qualidade Final do Produto = 90%

Para o julgamento do resultado da avaliação, compara-se o resultado com as faixas definidas, conforme ISO/IEC 9126, descrita a seguir:

() Excelente

- () Bom
- () Regular
- () Insuficiente

Para fins de comprovação, adotou-se os seguintes valores para as faixas:

Excelente	= 90,6% a 100%	= Aceito
Bom	= 75,6% a 90,5%	= Aceito
Regular	= 50,6% a 75,5%	= Necessita de verificação
Insuficiente	= 0% a 50,5%	= Rejeitado.

Exemplo:

Qualidade Final do Produto = 90%

Julgamento = Bom => Aceito

5.9 FUNCIONAMENTO DO SOFTWARE – DESCRIÇÃO DAS TELAS

Os passos e as atividades que o usuário/avaliador executa na avaliação do software, são descritos a seguir:

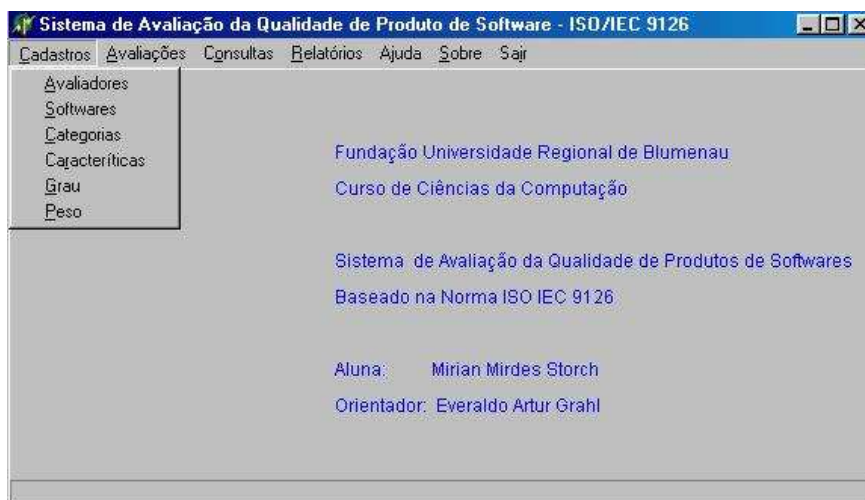
- a) o usuário/avaliador executa o protótipo e entra no menu principal, conforme a figura 12.

Nesta tela são apresentadas as seguintes opções:

- cadastros: Cadastro de Avaliadores, Cadastro de Softwares, Cadastro de Categorias de Software, Cadastro de Características, Subcaracterísticas e Perguntas, Cadastro de Graus de Importância das características e Cadastro de Pesos;
- avaliações: Configurar Avaliação;
- consultas: Consulta de todos os Cadastros , Consulta de Avaliações Realizadas;
- relatórios: Relatório Total e Relatório Parcial.

O usuário/avaliador escolhe a opção de acordo com o objetivo.

FIGURA 12 – TELA PRINCIPAL DO PROTÓTIPO



- b) se o usuário/avaliador deseja incluir avaliadores, ele deverá chamar a tela de cadastro de avaliadores, conforme a figura 13, onde deverão ser cadastrados todos os dados do avaliador, alterar dados cadastrados, ou excluir dados cadastrados;

FIGURA 13 – TELA DO CADASTRO DE AVALIADOR

Cadastro de Avaliadores

Identificação

Código: 1
Nome: Mirian Storch
Função: Avaliador

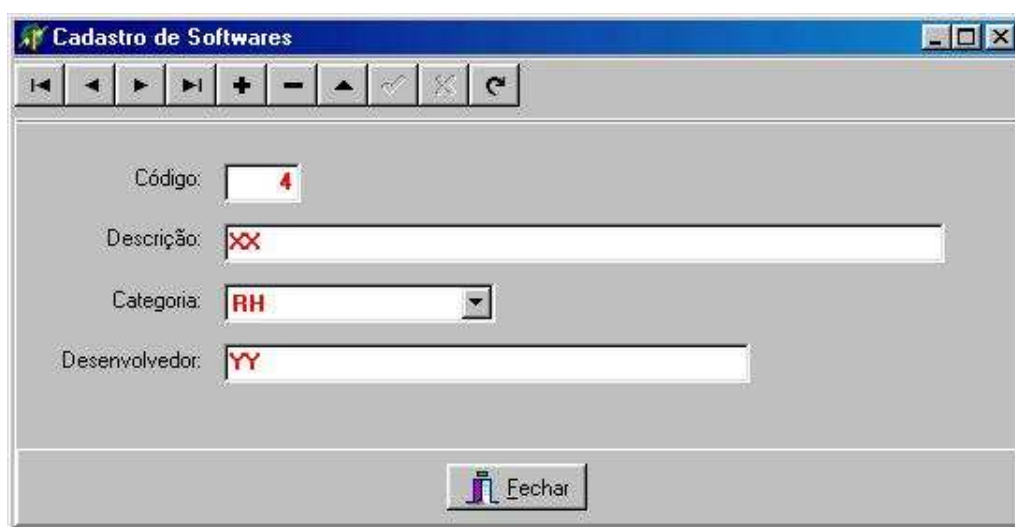
Endereço

Rua: Antonio da Veiga NR.: 555
Bairro: Vila Nova
Cidade: Blumenau
UF: SC
Telefone: 3400561

Fechar

- c) se o usuário/avaliador deseja incluir um software, ele deverá chamar a tela de Cadastro de Softwares, onde deverão ser cadastrados todos os dados do software a ser avaliado, alterar dados cadastrados, ou excluir dados cadastrados conforme a figura 14;

FIGURA 14 – TELA DO CADASTRO DE SOFTWARE



The screenshot shows a software registration window titled "Cadastro de Softwares". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Below the title bar is a toolbar with several icons: a left arrow, a right arrow, a plus sign, a minus sign, a checkmark, a cross, and a refresh symbol. The main content area contains four data entry fields. The first field is labeled "Código" and contains the number "4". The second field is labeled "Descrição" and contains "XX". The third field is labeled "Categoria" and is a dropdown menu currently showing "RH". The fourth field is labeled "Desenvolvedor" and contains "YY". At the bottom center of the window is a button labeled "Fechar" with a small icon to its left.

- d) o cadastramento de Categorias ao qual o software pertence, deve anteceder ao cadastramento de software, conforme a figura 15;

FIGURA 15 – TELA DO CADASTRO DE CATEGORIA



The screenshot shows a software category registration window titled "Cadastro de Categorias". The window has a standard Windows-style title bar with minimize, maximize, and close buttons. Below the title bar is a toolbar with several icons: a left arrow, a right arrow, a plus sign, a minus sign, a checkmark, a cross, and a refresh symbol. The main content area contains two data entry fields. The first field is labeled "Codigo" and contains the number "4". The second field is labeled "Descrição" and contains "RH". At the bottom center of the window is a button labeled "Fechar" with a small icon to its left.

- e) antes de iniciar uma avaliação, o usuário/avaliador deverá cadastrar as características, subcaracterísticas e perguntas que farão parte da avaliação,

conforme a figura 16. Nesta tela, o usuário poderá alterar os dados cadastrados, inserir novas características, subcaracterísticas e perguntas, ou excluir determinado dado que não se inclui na avaliação;

FIGURA 16 – TELA DO CADASTRO DE CARACTERÍSTICA

CodSubCaracteristica	NomeSubCaracteristica
11	Adequação
12	Acurácia
13	Interoperabilidade
14	Conformidade
15	Segurança

f) para cada avaliação, o usuário/avaliador deverá cadastrar ou atualizar os graus de importância de cada característica conforme a figura 17, e cadastrar ou alterar também os pesos de cada pergunta, conforme a figura 18;

FIGURA 17 – TELA DE CADASTRO DE GRAUS DE IMPORTÂNCIA

Codigo	Descrição	Valor
1	Alto	3

FIGURA 18 – TELA DE CADASTRO DE PESOS



Cadastro de Pesos

Codigo: 1

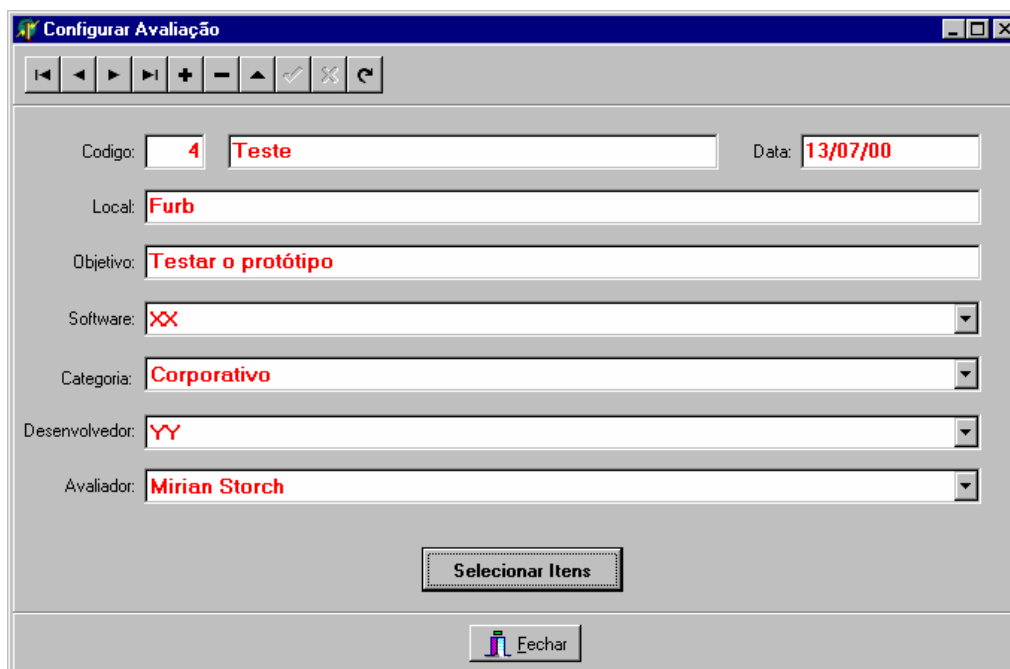
Descrição: Sim

Valor: 2

Fechar

g) após a entrada de todos os dados que farão parte da avaliação, o usuário/avaliador inicia o processo de configurar a avaliação, conforme a figura 19.

FIGURA 19 – TELA DE CONFIGURAR AVALIAÇÃO



Configurar Avaliação

Codigo: 4 Teste Data: 13/07/00

Local: Furb

Objetivo: Testar o protótipo

Software: XX

Categoria: Corporativo

Desenvolvedor: YY

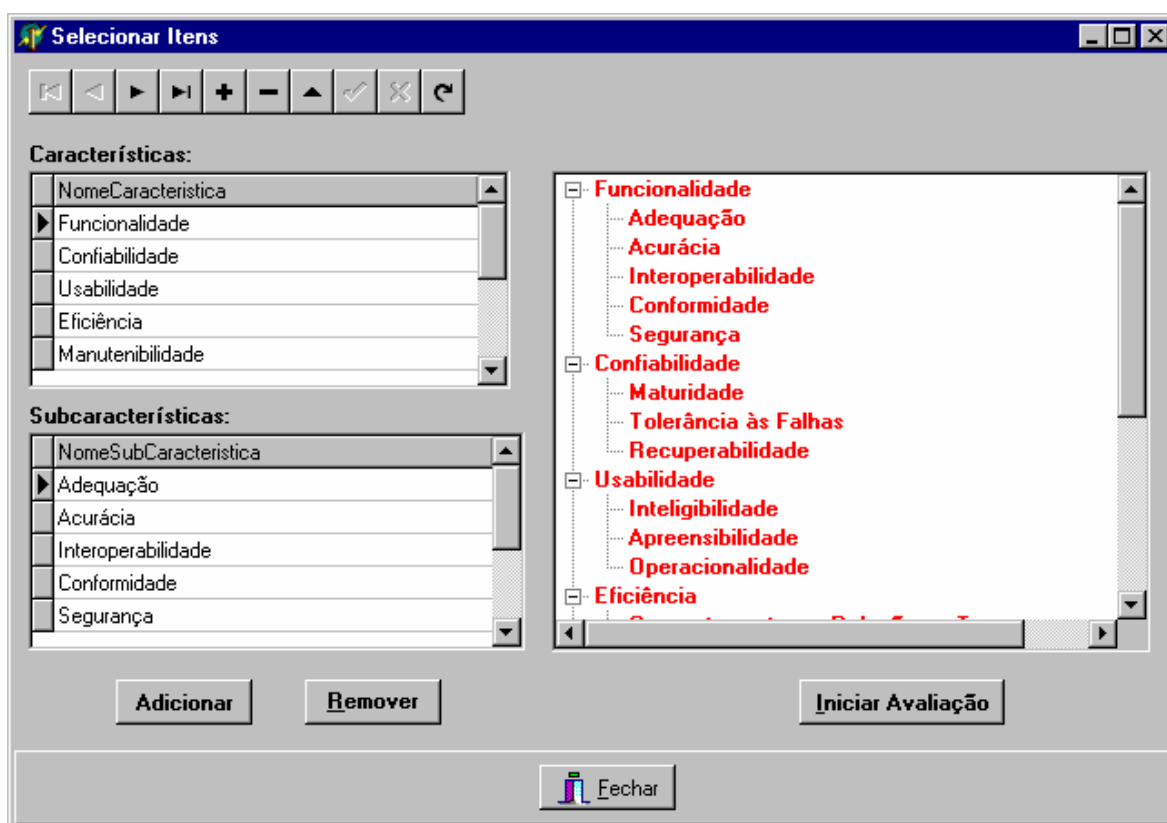
Avaliador: Mirian Storch

Selecionar Itens

Fechar

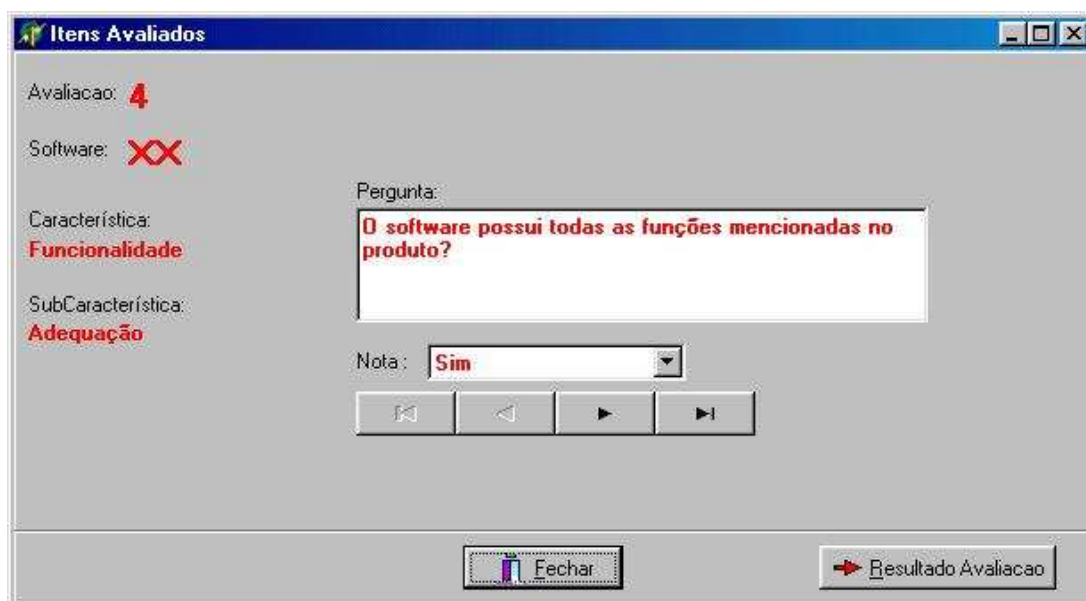
- h) o próximo passo é selecionar as características e subcaracterísticas que farão parte da avaliação, conforme a figura 20, e a partir daí, vai seguindo a avaliação passo-a-passo conforme a implementação do protótipo;

FIGURA 20 – TELA DE SELEÇÃO DAS CARACTERÍSTICAS



- i) após configurar uma avaliação, o usuário/avaliador entra no processo de avaliação das características, conforme a figura 21, onde ele irá navegar pelas características item após item, até chegar ao último item e das perguntas. Nesta tela, são apresentadas todas as perguntas cadastradas para as características e subcaracterísticas da qualidade, onde o usuário/avaliador irá responder o questionário dando uma nota a cada item avaliado, podendo retornar para a tela principal, ou seguir para o próximo passo, onde ele define o grau de importância para cada característica e obtém a média de cada característica e o resultado final da avaliação;

FIGURA 21 – TELA DE ITENS AVALIADOS



Itens Avaliados

Avaliacao: **4**

Software: **XX**

Característica: **Funcionalidade**

SubCaracterística: **Adequação**

Pergunta: **O software possui todas as funções mencionadas no produto?**

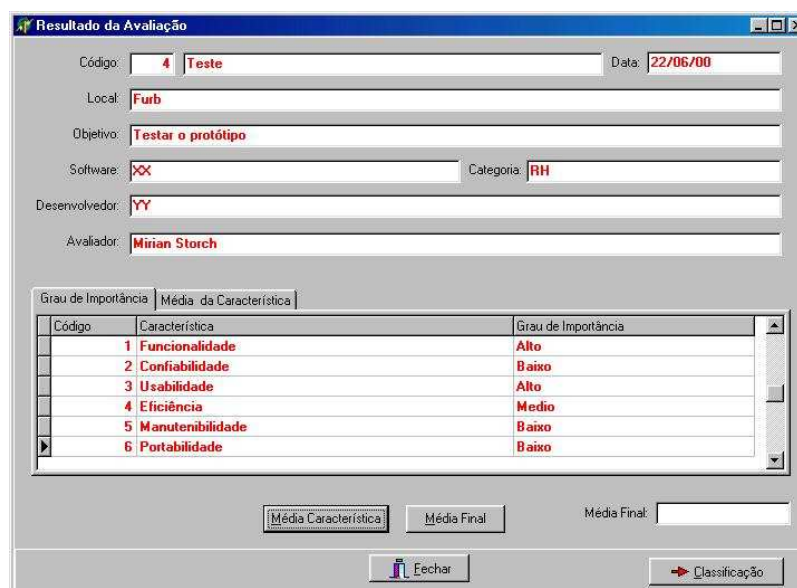
Nota: **Sim**

Fechar

Resultado Avaliação

- j) no passo seguinte, o usuário/avaliador define o grau de importância para cada característica conforme a figura 22, e a partir daí, o protótipo calcula a média de cada característica e o resultado final, apresentando o resultado da avaliação para cada característica da qualidade e a média geral da qualidade do software produto;

FIGURA 22 – TELA DE GRAU DE IMPORTÂNCIA DA CARACTERÍSTICA



Resultado da Avaliação

Código: **4** **Teste** Data: **22/06/00**

Local: **Furb**

Objetivo: **Testar o protótipo**

Software: **XX** Categoria: **RH**

Desenvolvedor: **YY**

Avaliador: **Mirian Storch**

Código	Característica	Grau de Importância
1	Funcionalidade	Alto
2	Confiabilidade	Baixo
3	Usabilidade	Alto
4	Eficiência	Medio
5	Manutenibilidade	Baixo
6	Portabilidade	Baixo

Média Característica

Média Final

Média Final:

Fechar

Classificação

FIGURA 23 – TELA DE RESULTADOS

Resultado da Avaliação

Código: Data:

Local:

Objetivo:

Software: Categoria:

Desenvolvedor:

Avaliador:

Código	Característica	Média Característica
1	Funcionalidade	01,000000
2	Confiabilidade	00,928571
3	Usabilidade	00,797872
4	Eficiência	01,000000
5	Manutenibilidade	01,000000
6	Portabilidade	00,940000

Média Característica Média Final Média Final:

k) durante o processo de avaliação, o usuário/avaliador poderá consultar determinado dado cadastrado, através das telas de consultas implementadas no protótipo. A figura 24 apresenta a tela de consulta de avaliadores;

FIGURA 24 – TELA DE CONSULTA DE AVALIADORES

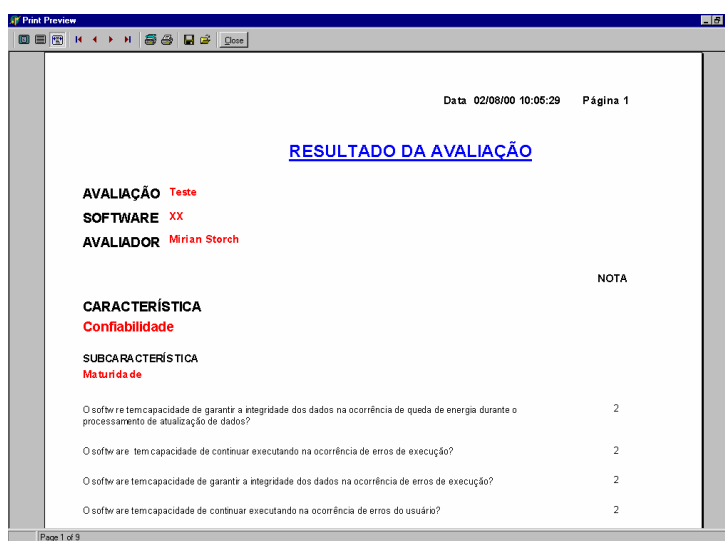
Consulta de Avaliadores

Digite o código ou nome do Avaliador

Código	Avaliador	Função	Rua	Nr	Bairro
1	Mirian Storch	Avaliador	Antonio da Veiga	555	Vila F
2	Everaldo Artur Grahl	Orientador do TCC	Antonio da Veiga	140	Vila F
3	Marcel Hugo	Professor	Antonio da veiga	140	Vila F
4	Carlos Negrão Bizzoto	Professor	Antonio da Veiga	140	Vila F
5	Mauricio C. Lopes	Professor	Antonio da Veiga	140	Vila F
6	José Roque Voltoline da Silva	Coordenador do TCC	Antônio da Veiga	140	Vila F

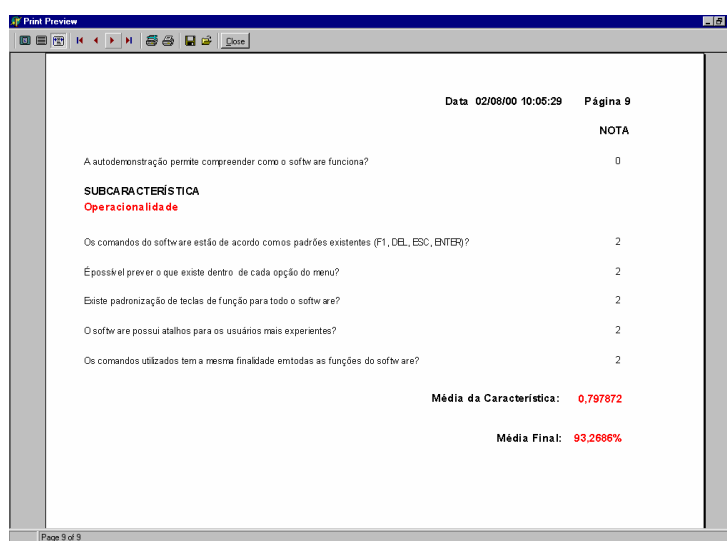
1) quando o usuário/avaliador tiver concluída a avaliação, ele poderá imprimir os resultados através dos relatórios implementados no Menu Principal. A figura 25 apresenta a tela da página inicial do relatório de resultados de uma avaliação, como as características, subcaracterísticas, perguntas, as notas dadas as perguntas, a média de cada característica e a média final do software avaliado. A figura 26 apresenta a tela da página final do relatório;

FIGURA 25 – TELA PÁGINA INICIAL DO RELATÓRIO DE RESULTADOS



CARACTERÍSTICA	NOTA
Confiabilidade	
SUBCARACTERÍSTICA	
Maturidade	
O software tem capacidade de garantir a integridade dos dados na ocorrência de queda de energia durante o processamento de atualização de dados?	2
O software tem capacidade de continuar executando na ocorrência de erros de execução?	2
O software tem capacidade de garantir a integridade dos dados na ocorrência de erros de execução?	2
O software tem capacidade de continuar executando na ocorrência de erros de usuário?	2

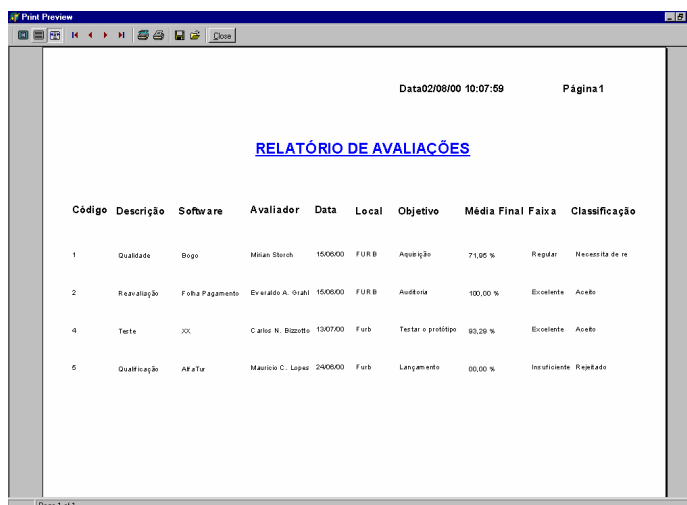
FIGURA 26 – TELA PÁGINA FINAL DO RELATÓRIO DE RESULTADOS



SUBCARACTERÍSTICA	NOTA
Operacionalidade	
A autodemonstração permite compreender como o software funciona?	0
Os comandos do software estão de acordo com os padrões existentes (F1, DEL, ESC, ENTER)?	2
É possível prever o que existe dentro de cada opção do menu?	2
Existe padronização de teclas de função para todo o software?	2
O software possui atalhos para os usuários mais experientes?	2
Os comandos utilizados tem a mesma finalidade em todas as funções do software?	2
Média da Característica:	0,797872
Média Final:	93,2686%

- m) o usuário/avaliador também poderá obter o relatório de todas as avaliações realizadas, contendo o software avaliado, o avaliador, a média final do software, a faixa em que ele se encaixa e sua classificação. A figura 27 apresenta a tela do relatório de todas as avaliações realizadas.

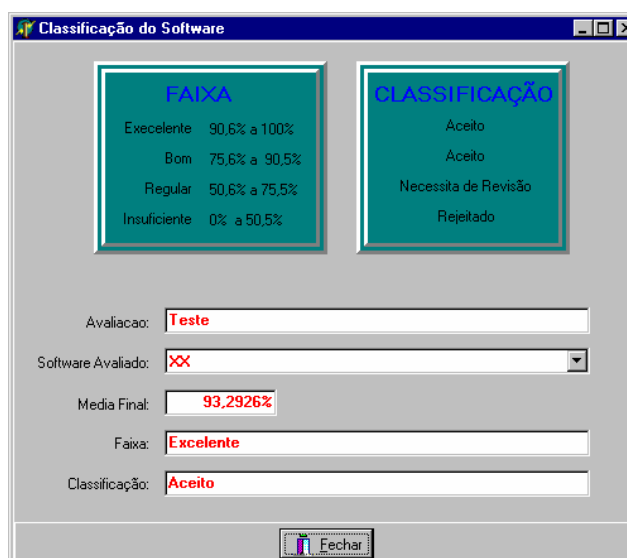
FIGURA 27 – TELA DO RELATÓRIO DE AVALIAÇÕES



Código	Descrição	Software	Avaliador	Data	Local	Objetivo	Média Final	Faixa	Classificação
1	Qualidade	Bogo	Milán Storck	15/06/00	FURB	Aquisição	71,05 %	Regular	Necessita de re
2	Reavaliação	Ficha Pagamento	Everaldo A. Orabi	15/06/00	FURB	Auditoria	100,00 %	Excelente	Aceito
4	Teste	XX	Carlos N. Bazzotto	13/07/00	Furb	Testar o protótipo	93,29 %	Excelente	Aceito
5	Qualificação	AfaTur	Maurício C. Lopes	24/05/00	Furb	Lançamento	00,00 %	Insuficiente	Rejeitado

- n) antes de encerrar a avaliação, o usuário/avaliador poderá verificar a faixa em que o software se encaixa e o julgamento conforme a média atingida. A figura 28 apresenta a tela de classificação do software.

FIGURA 28 – TELA DE CLASSIFICAÇÃO DO SOFTWARE



FAIXA

Excelente	90,6% a 100%
Bom	75,6% a 90,5%
Regular	50,6% a 75,5%
Insuficiente	0% a 50,5%

CLASSIFICAÇÃO

Aceito
Aceito
Necessita de Revisão
Rejeitado

Avaliação:

Software Avaliado:

Média Final:

Faixa:

Classificação:

5.10 ANÁLISE DO PROCESSO DE AVALIAÇÃO E DOS RESULTADOS OBTIDOS

Para a realização da avaliação foi necessário o levantamento dos requisitos da qualidade que deveriam estar presentes em um software de qualquer área. Baseado nesses requisitos, foi feito o detalhamento das características da qualidade descritos na norma e a partir daí, elaborou-se o questionário com as perguntas referente a cada subcaracterística e item de avaliação. A grande dificuldade na elaboração do questionário foi a formulação das perguntas para cada item de forma genérica, sob que ângulo avaliar cada item.

O processo da avaliação teste no protótipo foi realizado da seguinte forma:

- a) após o desenvolvimento e funcionamento do protótipo, foi realizado o cadastramento de todos os dados referente a avaliação, como por exemplo, os dados do avaliador, do software a ser avaliado, os pesos e graus utilizados, as características e subcaracterísticas e do questionário de avaliação dividido entre as características a serem avaliadas;
- b) após o cadastramento completo, iniciou-se o processo de avaliação, onde foi definida uma nota para cada pergunta do questionário. A nota foi dada de acordo com o atendimento do software em relação a cada item avaliado. Realizaram-se vários testes repetidos para algumas perguntas, para a confirmação dos resultados obtidos, visto que alguns itens, necessitam de um detalhamento maior. Na sequência, efetuou-se a execução dos cálculos para obtenção dos resultados.
- c) concluída a avaliação, foram gerados os relatórios com os resultados da avaliação teste;
- d) com base nos dados dos relatórios, verificou-se também que a maioria das respostas estavam relacionadas as opções Sim e Não, apenas algumas respostas tinham a opção Parcialmente. Isto demonstrou a necessidade de definir métricas específicas para determinados itens avaliados, como por exemplo, o número de itens atendidos;

- e) analisando os resultados, concluiu-se que o software avaliado, atende em 93% às necessidades do usuário, sendo classificado como Excelente;

- f) a experiência da avaliação teste realizada no protótipo, mostrou que existe uma grande dificuldade em se elaborar um modelo de avaliação, tanto no processo da montagem do modelo, quanto na execução da avaliação. Durante a avaliação pode-se perceber algumas falhas em itens implementados no software avaliado.

6 CONCLUSÃO

6.1 CONSIDERAÇÕES FINAIS

Este trabalho, procurou demonstrar a aplicação da Norma ISO/IEC 9126 que trata da qualidade de produto de software.

O objetivo do trabalho foi alcançado, visto que foi criado um roteiro de avaliação das principais características e subcaracterísticas da qualidade de forma genérica, que possa ser usado na avaliação de softwares em áreas diferentes.

O protótipo implementado é de fácil compreensão, permitindo que pessoas leigas possam utilizá-lo na realização de avaliação de qualquer produto de software, sendo necessário a ampliação do questionário, pois o protótipo apresenta de forma simplificada os requisitos da qualidade de um produto de software. O protótipo permite a manutenção das perguntas, tornando mais flexível a utilização deste roteiro de avaliação.

Para fins de teste do protótipo, realizou-se uma avaliação de um software na área de Recursos Humanos. O resultado da avaliação está descrito no Anexo B.

6.2 SUGESTÕES

Como sugestão para futuros trabalhos e aprimoramento do mesmo, torna-se necessário:

- a) ampliar o detalhamento das características e subcaracterísticas da qualidade da Norma ISO/IEC 9126;
- b) ampliar o questionário de avaliação do protótipo de forma genérica e de acordo com o tipo de software a ser avaliado;
- c) incluir o estudo da Norma ISO/IEC 14598-1 à ISO/IEC 14598-6 que trata da avaliação de produtos de softwares e da Norma ISO/IEC 12119 que trata dos requisitos da qualidade e testes.

ANEXO A

**QUESTIONÁRIO DE AVALIAÇÃO DAS CARACTERÍSTICAS DA
QUALIDADE DE PRODUTOS DE SOFTWARES
DE QUALQUER ÁREA**

CARACTERÍSTICA	SUB CARACTERÍSTICA	PERGUNTA	
1 Funcionalidade	1.1 Adequação	1.1.1 O software possui todas as funções mencionadas no produto?	
		1.1.2 O software dispõe de todas as funções necessárias para a sua execução?	
		1.1.3 O software dispõe de funções para processamento de rotinas específicas?	
	1.2 Acurácia	1.2.1 O software é preciso na execução das funções?	
		1.2.2 O software é preciso nos resultados?	
	1.3 Interoperabilidade	1.3.1 O software tem capacidade para processamento multiusuário?	
		1.3.2 O software tem capacidade para operação com redes?	
		1.3.3 O software tem restrições quanto ao número de estações trabalhando ao mesmo tempo?	
	1.4 Conformidade	1.4.1 O software é conciso às leis vigentes?	
	1.5 Segurança		1.5.1 O software informa ao usuário a necessidade da realização de <i>backups</i> na instalação de novas versões?
			1.5.2 O software dispõe de rotina interna de <i>backup</i> ?
			1.5.3 O software permite a compactação de <i>backups</i> ?
			1.5.4 O software dispõe segurança de acesso através de senhas?
			1.5.5 O software dispõe de rotina interna de <i>restore</i> ?

		1.5.6 O software possui controle de versão na restauração?
2 Confiabilidade	2.1 Maturidade	2.1.1 O software tem capacidade de continuar executando na ocorrência de erros de execução?
		2.1.2 O software tem capacidade de continuar executando na ocorrência de erros do usuário?
		2.1.3 O software tem capacidade de garantir a integridade dos dados na ocorrência de erros em execução?
		2.1.4 O software tem capacidade de garantir a integridade dos dados na ocorrência de queda de energia durante o processamento de atualização de dados?
	2.2 Tolerância à Falhas	2.2.1 O software possui advertência de erros cometidos pelo usuário?
		2.2.2 O software possui advertência quando ocorre erros na recuperação de arquivos?
		2.2.3 O software garante advertência quando ocorre erros de acesso ao software?
		2.2.4 O software controla preenchimento de campos?
		2.2.5 O software tem capacidade de controlar o preenchimento incorreto dos campos?
		2.2.6 O software tem capacidade de verificar se as informações cadastradas estão corretas?
		2.2.7 O software tem capacidade de evitar a inclusão de dados existentes?
		2.2.8 O software garante a integridade dos dados no caso de erros de execução?
		2.2.9 O software possui advertência de erros de configuração de impressora?
		2.2.10 O software permite mudar o modelo padrão de configuração do software?
		2.2.11 O software tem capacidade de voltar ao estado anterior após parada anormal da máquina?

		2.2.12 O software perde a integridade devido a paradas anormais?
		2.2.13 O software tem capacidade de informar ao usuário, a situação dos dados após paradas anormais?
		2.2.14 O software tem capacidade de continuar o processamento com grande volume de dados?
		2.2.15 O software tem capacidade de suportar usuários simultâneos?
	2.3 Recuperabilidade	2.3.1 O software tem capacidade de alterar senhas incorretas?
		2.3.2 O software possui arquivo temporário para evitar a perda de dados no caso de desligamento do equipamento sem salvar as últimas alterações?
		2.3.3 O software tem capacidade de recuperar dados excluídos no caso de erros de execução?
		2.3.4 Em caso de erros de execução ocorre erros de informação?
		2.3.5 O software permite recuperar dados excluídos em caso de erros de software?
		2.3.6 O software tem capacidade de recuperar dados excluídos?
		2.3.7 A rotina de recuperação do software é fácil de ser usada?
		2.3.8 A recuperação dos dados ocorre de forma rápida?
3 Usabilidade	3.1 Inteligibilidade	3.1.1 O software possui autodemonstração?
		3.1.2 A autodemonstração permite compreender como o software funciona?
		3.1.3 A autodemonstração é facilmente acessada?
		3.1.4 O software possui tutorial <i>on-line</i> ?
		3.1.5 Após a instalação, o software oferece a possibilidade de usar o tutorial?
		3.1.6 As telas do software são autoinstrutivas, permitindo ao usuário visualizar com facilidade qual sua função?
		3.1.7 As telas do mesmo nível possui o mesmo padrão?

		3.1.8 Os itens de menus possuem termos lógicos de fácil compreensão?
		3.1.9 Os itens de menus são padronizados, possuindo sempre o mesmo significado?
		3.1.10 A ordem de apresentação dos menus segue uma lógica?
		3.1.11 Os submenus mantêm a mesma lógica dos menus?
		3.1.12 O usuário precisa ter profundo conhecimento na área para utilizar o software?
		3.1.13 Caso o usuário seja leigo, o software fornece as informações adequadas para sua perfeita utilização?
		3.1.14 A teoria que embasa o software é explicada com documentação impressa?
		3.1.15 Os jargões técnicos utilizados, são explicados no manual do usuário?
	3.2 Apreensibilidade	3.2.1 O software possui manual de instalação ?
		3.2.2 O software possui manual de operação?
		3.2.3 O manual de instalação possui índice Analítico?
		3.2.4 O manual de instalação do software possui índice Remissivo?
		3.2.5 O manual de operação do software possui índice Analítico?
		3.2.6 O manual de operação do software possui índice Remissivo?
		3.2.7 O manual de operação do software possui índice de figuras?
		3.2.8 O manual de operação do software inclui os termos técnicos utilizados?
		3.2.9 Existe uma hierarquia de manuais de acordo com o nível de conhecimento do usuário?
		3.2.10 Os termos são usados com o mesmo significado durante todo o processamento?
		3.2.11 Todas as funções do software estão explicadas no manual?
		3.2.12 Todos os passos para a instalação do software estão claramente apresentados?

		3.2.13 A configuração de hardware mínima para a instalação estão claramente definidas?
		3.2.14 Os possíveis erros de instalação são apresentados claramente no manual?
		3.2.15 O manual apresenta exemplos de como utilizar o software?
		3.2.16 O manual apresenta todos os erros que podem ocorrer no software?
		3.2.17 Os textos dos manuais são corretamente escritos?
		3.2.18 O manual do usuário explica as convenções de estilo utilizadas durante o documento (negrito, itálico, etc.)?
		3.2.19 O volume de texto está de acordo com a quantidade de informações obtidas?
		3.2.20 O software possui ajuda <i>on-line</i> ?
		3.2.21 O software possui ajuda <i>on-line</i> de como corrigir os erros cometidos pelo usuário?
		3.2.22 A ajuda <i>on-line</i> possui um índice?
		3.2.23 Os termos utilizados tem o mesmo significado em todo arquivo de ajuda?
		3.2.24 As mensagens de ajuda apresentam uma explicação para todos os itens relacionados à utilização do software?
		3.2.25 A linguagem utilizada na mensagem de ajuda é facilmente entendida pelo usuário?
		3.2.26 O software mostra com o usuário deve navegar pelo arquivo de ajuda?
		3.2.27 As mensagens de orientação estão padronizadas?
	3.3 Operacionalidade	3.3.1 Os comandos utilizados tem a mesma finalidade em todas as funções do software?
		3.3.2 Os comandos do software estão de acordo com os padrões existentes (F1, DEL, ESC, ENTER)?
		3.3.3 É possível prever o que existe dentro de cada opção do menu?

		3.3.4 Existe padronização de teclas de função para todo o software?
		3.3.5 O software possui atalhos para os usuários mais experientes?
4 Eficiência	4.1 Comportamento em Relação ao Tempo	4.1.1 O tempo necessário para a instalação do software é satisfatório?
		4.1.2 O tempo necessário para inicializar o software é satisfatório?
		4.1.3 O tempo necessário para fechar o software é satisfatório?
		4.1.4 O tempo de resposta é adequado em relação do volume de dados envolvidos?
		4.1.5 O tempo de resposta é adequado a complexidade das funções do software?
		4.1.6 O tempo de resposta para a realização de consultas é satisfatório?
		4.1.7 O tempo necessário para a realização de <i>backups</i> é satisfatório?
	4.2 Utilização dos Recursos	4.2.1 Os recursos do equipamento exigidos pelo software são adequados a complexidade das funções?
		4.2.2 O acesso a disco no software está de acordo com a complexidade das funções durante a configuração do equipamento usado?
		4.2.3 O software é coerente na utilização da memória expandida?
		4.2.4 O software faz muito acesso a disco?
		4.2.5 O software permite a sua operação durante a impressão de documentos?
5 Manutenibilidade	5.1 Analisabilidade	5.1.1 O software contém uma saída para cada entrada?
		5.1.2 O software contém funções com objetivos específicos?
		5.1.3 O software utiliza padrão para nome de identificadores?

		5.1.4 O software utiliza nomes significativos e concisão para os indicadores?
		5.1.5 O software utiliza somente variáveis locais, evita as globais?
		5.1.6 O software possui funções com objetivos específicos?
		5.1.7 O software possui as decisões comentadas?
		5.1.8 O software possui as variáveis são descritas por comentários?
		5.1.9 O software possui os desvios comentados?
		5.1.10 O software possui toda a programação em linguagem de máquina comentada?
		5.1.11 O nome de todas as variáveis do software são exclusivos?
		5.1.12 As variáveis são usadas apenas de uma única forma?
		5.1.13 As variáveis globais são usadas consistentemente em relação à unidade e tipos?
		5.1.14 Todas as variáveis são inicializadas antes do uso?
		5.1.15 Todos os valores de default são descritos?
		5.1.16 O software tem capacidade de verificar as entradas
		5.1.17 As variáveis locais são definidas?
		5.1.18 O software evita o código automodificável?
		5.1.19 O software verifica possíveis conflitos ou combinações legais de entradas?
	5.2 Modificabilidade	5.2.1 O software possui notação padronizada para descrever interfaces?
		5.2.2 O software possui notação padronizada para descrever estrutura de dados?
		5.2.3 O Software permite alterações para acomodar um novo protocolo de comunicação?
		5.2.4 O software permite modificações para ser usado em uma máquina diferente?
		5.2.5 O software permite alterações para adicionar um novo drive?

		5.2.6 O software possui documentação técnica legível?
		5.2.7 O software possui Dicionário de Dados bem estruturado para facilitar a modificação?
	5.3 Estabilidade	5.3.1 O software tem capacidade de evitar a necessidade de manutenção na ocorrência de erros?
		5.3.2 O software tem capacidade de evitar a atualização de versões frequentes?
		5.3.3 O software consegue evitar erros após a manutenção dos mesmos?
		5.3.4 O software tem capacidade de executar a manutenção com rapidez?
		5.3.5 O software utiliza técnicas de encapsulação da informação?
	5.4 Testabilidade	5.4.1 O software possui uma base de demonstração para realização de testes?
		5.4.2 O software tem capacidade de executar automaticamente os testes para a validação das modificações?
		5.4.3 O software possui um guia de testes?
		5.4.4 O software possui documentação de testes e configuração do software?
		5.4.5 O software especifica fundamentos para cada caso de teste?
		5.4.6 O software especifica uma descrição dos resultados esperados em cada teste?
6 Portabilidade	6.1 Adaptabilidade	6.1.1 O software pode ser facilmente modificado para atender às necessidades do usuário?
		6.1.2 O software possui versão para utilizar em rede?
		6.1.3 O software tem capacidade para operar em ambientes diferentes?
		6.1.4 O software pode ser facilmente modificado para atender as alterações sugeridas pelo usuário?
		6.1.5 O software possui rotinas para configuração de drivers e impressoras?
		6.1.6 O software permite adicionar funções com facilidade?

		6.1.7 O software permite deletar funções com facilidade?
	6.2 Instalação	6.2.1 O software possui um programa de instalação?
		6.2.2 Os comandos utilizados durante a instalação são de fácil entendimento?
		6.2.3 O software possui <i>help</i> de instalação?
		6.2.4 O software possui uma ordem lógica na seqüência dos disquetes de instalação?
		6.2.5 O software possui alguma indicação no andamento da instalação?
		6.2.6 O software possui uma demonstração do software enquanto o usuário instala o mesmo?
		6.2.7 O software faz a instalação sem a intervenção do usuário?
		6.2.8 O software tem capacidade de realizar instalação compactada?
		6.2.9 O software informa o usuário da necessidade de fazer <i>backup</i> antes de fazer a instalação da nova versão?
		9.2.10 O software informa o usuário das alterações que serão realizadas na configuração do equipamento?
		6.2.11 O software verifica se há espaço disponível para a instalação?
		6.2.12 O software é autosugestivo nas opções de instalações?
	6.3 Conformidade	6.3.1 O software tem capacidade de ser utilizado em diferentes tipos de hardware e com diferentes configurações?

		6.3.2 O software tem capacidade de ser utilizado independente da versão de sistema operacional existente?
		6.3.3 O software possui interface ODBC?
	6.4 Substituição	6.4.1 O software tem capacidade de ser substituído por novas versões e continuar utilizando a mesma base de dados?
		6.4.2 O software tem capacidade de continuar funcionando sem sofrer modificações quando da troca de ambiente?
		6.4.3 O software tem condições de executar todas as funções necessárias?

ANEXO B

EXEMPLO DE AVALIAÇÃO DE PRODUTO DE SOFTWARE

O relatório contendo o resultado da avaliação exemplo realizada no protótipo, encontra-se na página a seguir.

GLOSSÁRIO

Banco de dados	- coleção de dados armazenados eletronicamente
Configuração	- capacidade de um sistema para sofrer mudança ou personalização
Desenvolvedor	- pessoa que desenvolve software
Diagrama	- representação gráfica de programas
FIRMWARE	- hardware que contém dados e um programa de computador que não pode ser alterado no ambiente do usuário
Fluxograma	- representação do fluxo e entradas e saídas de programa
Hardware	- máquinas e equipamentos quando em operação fazem armazenamento e transmissão de dados e programas
Interface	- conexão e interação entre hardware e software e o usuário
Julgamento	- ação de aplicar critérios de julgamento específicos e documentados a um produto, com o propósito de determinar sua aplicação
Medição	- ação de aplicar uma métrica de qualidade de software a um produto de software específico
Menu	- lista na tela das funções ou operações disponíveis que podem ser efetuadas atualmente
Métrica	- valor medido
Pontuação	- ação de mapear o valor medido ao nível de pontuação apropriada
Runtime	- tempo de execução, refere-se a execução real de um programa
Software	- programa de computador
SCRIPTS	- relatório
WINDOWS	- sistema operacional baseado em gráficos.

7 REFERÊNCIAS BIBLIOGRÁFICAS

- [BAR1997] BARRETO, José Carlos. **Qualidade de software**. Endereço Eletrônico : www.bareto.com.br/qualidade, 1997.
- [CAM1992] CAMPOS, Vicente Falconi. Controle da qualidade total. Belo Horizonte : Bloch Editores, 1992.
- [CHO1999] CHOOSE Technologies, **Roteiro para Avaliação 2001**. São Paulo, 1999.
- [ELI1999] ELIAS, Amira Al Farah, Vostoupal, Tânia Mara. **Núcleo de avaliação de produtos de software**. Endereço eletrônico: www.pr.gov.br/celepar/celepar/batebyte/bb81/ava, 1999.
- [FER1995] FERNANDES, Aguinaldo Aragon . Gerência de software através de métricas: garantindo a qualidade do projeto, processo e produto. São Paulo : Atlas, 1995.
- [FIS1998] FISHER, Alan S. **Case – utilização de ferramentas para desenvolvimento de software**. Rio de Janeiro : Campus, 1998.
- [GIL1995] GILLIES, Alan C. **Software quality – teory and management**. Londres : Chapman & Hall, 1995.
- [GUE1995] GUESSER, Enezilda. **Protótipo para avaliação da qualidade de softwares de automação comercial**. Trabalho de Conclusão de Curso do Centro de Ciências Exatas e Naturais. Universidade Regional de Blumenau, 1995.
- [ISO1993] International Organization For Standardization, International Eleetrotechnical Commission. **Software product evolution – Genera Guide**. NO. 1136, Canadá, julho de 1993.

- [ISO1995] International Organization For Standardization, International Elettrotechnical Commission. **SC7/WG6 Project 7-13 Product Requeriments**, ISO/IEC 9126. Japan, 1995.
- [ISO1996] International Organization For Standardization, International Elettrotechnical Commission. **Information Tecnology – Software Product Evaluation** – Quality characteristic and guidlines for theis use. Irlanda, 1996.
- [MIL1994] MILL, Charles A. **Auditoria da qualidade** . São Paulo : Makron Books, 1994.
- [NOB1994] NÓBREGA, Kleber Cavalcanti. **Apostila do curso de gestão da qualidade - especialização em gestão da qualidade**. Blumenau, 1994.
- [PIS1996] PISKE, Rosilene. **Protótipo para avaliação da qualidade de softwares de folha de pagamento**. Trabalho de Conclusão de Curso do Centro de Ciências Exatas e Naturais. Universidade Regional de Blumenau, 1996.
- [POF1995] POFFO, Márcio A. **Protótipo para avaliação da qualidade de softwares de contabilidade**. Trabalho de Conclusão de Curso do Centro de Ciências Exatas e Naturais. Universidade Regional de Blumenau, 1995.
- [PUR1995] PURI, Subhash C. **ISO 9000 Certification – total quality management**. Ontário : Vision, 1995.
- [ROC1995] ROCHA, Ana Regina Cavalcanti da. **Controle da qualidade de software**. Coordenação de programas de pós-graduação em engenharia. Programa de engenharia de sistemas e computação, Universidade Federal do Rio de Janeiro, COPPE, 1995.
- [SAN1998] SANDRI, Vivian. **Software de apoio a avaliação da qualidade de pacotes basedao na norma ISO/IEC 12119**. Trabalho de Conclusão de Curso do Centro de Ciências Exatas e Naturais. Universidade Regional de Blumenau, 1998.

- [SHI1992] SHILLER, Lary. **Excelência em software**. São Paulo : Makron Books, 1992 .
- [TSU1997] TSUKUMO, Claudete M. Rêgo, et. **Avaliação de Produtos de Software:** algumas questões relevantes. Anais do IX Simpósio Brasileiro de Engenharia de Software – SBES 95. Recife, 1997.
- [WEB1997] WEBER, Kital Chaves. **Qualidade e produtividade em software**. São Paulo : Makron Books, 1997.
- [WEI1994] WEINBERG, Gerald M. **Software com qualidade**. São Paulo : Makron Books, 1994.