

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

**FERRAMENTA PARA AVALIAÇÃO DA QUALIDADE DE  
PRODUTOS DE SOFTWARE BASEADO NAS NORMAS  
ISO/IEC 9126 E NBR ISO/IEC 12119**

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

**FABIANO STIMAMIGLIO FERREIRA**

BLUMENAU, JULHO/2003

2003/1-25

# **FERRAMENTA PARA AVALIAÇÃO DA QUALIDADE DE PRODUTOS DE SOFTWARE BASEADO NAS NORMAS ISO/IEC 9126 E NBR ISO/IEC 12119**

**FABIANO STIMAMIGLIO FERREIRA**

ESTE TRABALHO DE CONCLUSÃO DE CURSO FOI JULGADO ADEQUADO  
PARA OBTENÇÃO DOS CRÉDITOS NA DISCIPLINA DE TRABALHO DE  
CONCLUSÃO DE CURSO OBRIGATÓRIA PARA OBTENÇÃO DO TÍTULO DE:

**BACHAREL EM CIÊNCIAS DA COMPUTAÇÃO**

---

Prof. Carlos Eduardo Negrão Bizzoto — Orientador na FURB

---

Prof. José Roque Voltolini da Silva — Coordenador do TCC

## **BANCA EXAMINADORA**

---

Prof. Carlos Eduardo Negrão Bizzotto

---

Prof. Everaldo Artur Grahl

---

Prof. Ricardo Alencar Azambuja

## DEDICATÓRIA

Aos meus pais, por acreditarem em mim indissolavelmente, sempre me apoiando até a vitória final. Amo vocês.

A minha amada esposa, Alexandra, pela extrema compreensão e apoio no decorrer deste trabalho e em todos os momentos de nossas vidas.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus, pelo dom da vida, por ter me dado esta oportunidade de aprendizado e principalmente pela família maravilhosa com a qual me abençoou.

Aos meus queridos pais Sergio e Teresinha, que me educaram, me apoiaram e sempre estiveram ao meu lado quando precisei de um ombro amigo.

A minha adorada irmã Karen, pelo incentivo e alegria de viver.

Ao meu querido irmão Sergio, que mesmo distante sempre me apoiou.

Ao amigo Jean Fábio Fuchs, pelo incentivo e apoio proporcionado durante a elaboração deste trabalho.

Aos meus amigos que compreenderam a minha ausência.

Ao orientador Carlos Eduardo Negrão Bizzotto, pela atenção e auxílio dispensados na elaboração deste trabalho.

E a todos os que me estimam e que me querem bem.

## **RESUMO**

Este trabalho visa apresentar um estudo sobre as normas ISO/IEC 9126 e NBR ISO/IEC 12119. Além disso, propõe uma forma para a avaliação da Qualidade de Produtos de Software baseado nestas normas. Para tornar a avaliação menos subjetiva, faz-se um detalhamento das características e subcaracterísticas destas normas. Um aplicativo 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 and NBR ISO/IEC 12119 standards. Besides, this project proposes a way to evaluate the Quality of Software Products, based on these standards. Purposing to change the evaluation less subjective, it's necessary to detail the characteristics and sub-characteristics from these standards. An applicative for software quality evaluation was implemented to support the evaluator during the process of evaluation.

## LISTA DE FIGURAS

Figura 3-1 - Processo de avaliação .....	27
Figura 3-2 - Níveis de pontuação para as métricas.....	30
Figura 3-3 – Estrutura da norma ISO/IEC 12119.....	37
Figura 4-1 - Diagrama de contexto.....	51
Figura 4-2 - Diagrama de fluxo de dados.....	52
Figura 4-3 - Modelo entidade relacionamento .....	54
Figura 4-4 - Tela principal do sistema.....	66
Figura 4-5 – Tela de cadastro de avaliadores .....	68
Figura 4-6 – Tela de cadastro de produtoras .....	69
Figura 4-7 – Tela de cadastro de softwares .....	69
Figura 4-8 – Tela de cadastro de categorias .....	70
Figura 4-9 – Tela de cadastro de características.....	71
Figura 4-10 – Tela de cadastro das questões .....	71
Figura 4-11 – Tela dos tipos de avaliação .....	72
Figura 4-12 – Tela inicial da avaliação .....	73
Figura 4-13 – Movimentação da avaliação.....	74
Figura 4-14 – Tela de resultado da avaliação .....	75

## LISTA DE QUADROS

Quadro 2-1 - Fatores explícitos .....	18
Quadro 2-2 – Fatores implícitos .....	18
Quadro 2-3 – Normas nacionais e internacionais na área de software .....	24
Quadro 3-1 - Requisitos de qualidade para documentação de usuário .....	41
Quadro 4-1 – Lista de trabalhos correlatos.....	49
Quadro 4-2 - Entidade de avaliações (dicionário de dados).....	56
Quadro 4-3 - Entidade de avaliadores (dicionário de dados) .....	57
Quadro 4-4 - Entidade de características (dicionário de dados).....	57
Quadro 4-5 - Entidade de categorias (dicionário de dados) .....	57
Quadro 4-6 - Entidade de movimentos de avaliação (dicionário de dados).....	58
Quadro 4-7 - Entidade de movimento dos tipos de avaliação (dicionário de dados).....	58
Quadro 4-8 - Entidade de questões (dicionário de dados).....	58
Quadro 4-9 - Entidade de produtoras (dicionário de dados) .....	58
Quadro 4-10 - Entidade de resultado da avaliação (dicionário de dados).....	59
Quadro 4-11 - Entidade de tipos de avaliações (dicionário de dados) .....	59
Quadro 4-12 - Entidade de softwares (dicionário de dados) .....	60
Quadro 4-13 - Resultado da Avaliação .....	65



## LISTA DE ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
CASE	<i>Computer Aided Systems Enginneering</i>
DER	Diagrama Entidade Relacionamento
IEC	<i>International Eletrotechnical Committe</i>
ISO	<i>International Organization for Standardization</i>
NBR	Norma Brasileira de Regulamentação

# SUMÁRIO

1	INTRODUÇÃO .....	13
1.1	OBJETIVOS DO TRABALHO .....	14
1.2	RELEVÂNCIA DO TRABALHO .....	14
1.3	ESTRUTURA DO TRABALHO .....	15
2	QUALIDADE .....	16
2.1	CONCEITO DE QUALIDADE .....	16
2.2	QUALIDADE SOFTWARE .....	17
2.2.1	IMPORTÂNCIA DA QUALIDADE DE SOFTWARE .....	19
2.2.2	O QUE É UM SISTEMA DA QUALIDADE? .....	20
2.2.3	VISÕES DA QUALIDADE DE SOFTWARE .....	21
2.2.3.1	visão do usuário .....	21
2.2.3.2	visão da equipe de desenvolvimento .....	21
2.2.3.3	visão do gerente .....	22
2.3	QUALIDADE DE PRODUTOS DE SOFTWARE .....	22
2.4	NORMAS DE QUALIDADE DOS PRODUTOS DE SOFTWARE .....	23
3	NORMAS ISO/IEC 9126 E NBR ISO/IEC 12119 .....	26
3.1	NORMA ISO/IEC 9126 (NBR 13596) .....	26
3.1.1	PROCESSO DE AVALIAÇÃO DE PRODUTOS DE SOFTWARE .....	26
3.1.1.1	Estabelecer requisitos de avaliação .....	28
3.1.1.2	Especificar a avaliação .....	28
3.1.1.3	projetar a avaliação .....	29
3.1.1.4	executar a avaliação .....	29
3.1.2	MODELO DA QUALIDADE DEFINIDO PELA NORMA ISO/IEC 9126 .....	30

3.1.2.1 modelo para características externas e internas .....	31
3.1.2.1.1 funcionalidade .....	31
3.1.2.1.2 confiabilidade .....	31
3.1.2.1.3 usabilidade .....	32
3.1.2.1.4 eficiência .....	32
3.1.2.1.5 manutenibilidade .....	33
3.1.2.1.6 portabilidade .....	33
3.1.2.2 modelo para qualidade em uso .....	35
3.2 NORMA NBR ISO/IEC 12119 .....	36
3.2.1 REQUISITOS DA QUALIDADE PARA DESCRIÇÃO DE PRODUTO .....	38
3.2.1.1 requisitos gerais sobre o conteúdo da descrição .....	38
3.2.1.2 identificações e indicações .....	38
3.2.1.3 declarações sobre funcionalidade .....	39
3.2.1.4 declarações sobre confiabilidade .....	39
3.2.1.5 declarações sobre usabilidade .....	40
3.2.1.6 declarações sobre eficiência .....	40
3.2.1.7 declarações sobre manutenibilidade .....	41
3.2.1.8 declarações sobre portabilidade .....	41
3.2.2 REQUISITOS DE QUALIDADE PARA DOCUMENTAÇÃO DE USUÁRIO .....	41
3.2.3 REQUISITOS DE QUALIDADE PARA PROGRAMAS E DADOS .....	42
3.2.3.1 funcionalidade .....	43
3.2.3.2 confiabilidade .....	43
3.2.3.3 usabilidade .....	44
3.2.3.4 eficiência .....	45
3.2.3.5 manutenibilidade .....	45

3.2.3.6 portabilidade .....	45
3.2.4 INSTRUÇÕES PARA TESTE .....	46
4 DESENVOLVIMENTO DO TRABALHO .....	48
4.1 ANÁLISE DOS REQUISITOS.....	48
4.2 ESPECIFICAÇÃO .....	50
4.2.1 DIAGRAMA DE CONTEXTO.....	50
4.2.2 DIAGRAMA DE FLUXO DE DADOS (DFD) .....	51
4.2.3 DIAGRAMA ENTIDADE RELACIONAMENTO (DER).....	54
4.2.4 DICIONÁRIO DE DADOS .....	55
4.3 IMPLEMENTAÇÃO .....	60
4.3.1 ESTUDO DE CASO.....	62
4.3.2 DESCRIÇÃO DAS TELAS DA FERRAMENTA.....	66
4.4 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO .....	76
5 CONCLUSÕES .....	77
5.1 SUGESTÕES .....	78
REFERÊNCIAS BIBLIOGRÁFICAS .....	80
APÊNDICE A – LISTAGEM DO <i>CHECKLIST</i> DESENVOLVIDO A PARTIR DAS NORMAS ISO/IEC 9126 E NBR ISO/IEC 12119.....	83
APÊNDICE B – RELATÓRIO DE AVALIAÇÃO DO PRODUTO DE SOFTWARE .....	<u>101</u>
APÊNDICE C – RELATÓRIO DE COMPARAÇÃO DE AVALIAÇÃO DE DOIS PRODUTOS DE SOFTWARE .....	<u>102</u>
APÊNDICE D – RELATÓRIO DE AVALIAÇÃO EM BRANCO .....	<u>103</u>

Excluído: 101

Excluído: 101

Excluído: 102

Excluído: 102

Excluído: 103

Excluído: 103

# 1 INTRODUÇÃO

A partir do momento que se passa a ter uma economia globalizada, a concorrência entre as empresas que já era grande, passa a ser ainda mais acirrada, uma vez que a competição ocorre não só com os concorrentes locais ou regionais, mas também com empresas estrangeiras (Poffo, 1995).

Em função dessa realidade, as empresas têm procurado aumentar a qualidade de seus produtos, como forma de diferenciação com relação à concorrência. Isso tem sido realizado, dentre outras ações, através da implantação de processos mais racionais e produtivos na área de desenvolvimento de software, principalmente através da utilização das Tecnologias da Informação e da Comunicação. Essa difusão da informática, apesar de ter contribuído para o aumento da produtividade, tem causado, em alguns casos, diversos problemas relacionados à segurança, confiabilidade e interface com o usuário. Diante destes problemas, as empresas que desenvolvem software devem procurar associar sua imagem a produtos com qualidade. Para isto torna-se necessário a definição de padrões de qualidade, com relação aos quais os produtos de software possam ser validados.

Qualidade de software é um tema que está em evidência devido à preocupação com os custos na manutenção de software e à baixa conformidade nos requisitos dos mesmos. Para isso a *International Organization for Standardization (ISO)* e o *International Electrotechnical Commission (IEC)*, publicaram em 1991, a norma que representa a atual padronização mundial para qualidade de produtos de software denominada ISO/IEC 9126, que é aplicada para fazer a verificação da qualidade de produtos de software. Esta norma representa um dos primeiros passos no sentido de se definir padrões da qualidade de software onde, além disso, com a continuidade das pesquisas, a ISO/IEC 9126 deve evoluir no sentido de realizar a certificação de produtos de software. Para que isto seja possível, torna-se essencial o seu detalhamento em níveis mais específicos, onde o grau de subjetividade com relação à avaliação seja minimizado.

Em conjunto com a norma ISO/IEC 9126, surge em 1994 a norma ISO/IEC 12119, que traduzida e validada pela Associação Brasileira de Normas Técnicas em 1998, passou a chamar NBR ISO/IEC 12119, e tem como objetivo verificar e validar pacotes de software que é o conjunto completo e documentado de programas fornecidos a diversos usuários para uma

aplicação ou função genérica tais como: processadores de texto, planilhas eletrônicas, banco de dados, software gráficos, programas para funções técnicas ou científicas e programas utilitários (Sandri, 1998).

As normas ISO/IEC 9126 e NBR ISO/IEC 12119 tem por objetivo estabelecer requisitos e instruções de qualidade a respeito de como testar um pacote de software em relação aos requisitos estabelecidos para o pacote. Para que seja feita a aplicação correta destas normas, as mesmas deverão especificar os requisitos de qualidade de pacotes de software e estabelecer um esquema de avaliação ou certificação.

## 1.1 OBJETIVOS DO TRABALHO

O objetivo do presente trabalho é especificar e implementar uma ferramenta para auxiliar na avaliação da qualidade de produtos de software.

Os objetivos específicos do trabalho são:

- a) elaboração de um *checklist* baseado nas normas ISO/IEC 9126 e NBR ISO/IEC 12119;
- b) desenvolvimento de uma ferramenta de apoio ao processo de avaliação da qualidade de produtos de software baseado nas normas ISO/IEC 9126 e NBR ISO/IEC 12119, para que o usuário possa ter condições de escolher dentre os diversos softwares existentes, aquele que melhor atende as suas necessidades;
- c) elaborar um modelo de qualidade para produtos de software (*checklist*) para que desenvolvedores de sistemas possam ter um ponto de referência em termos dos requisitos que devem ser satisfeitos por seus produtos;
- d) fazer com que o usuário/avaliador tenha possibilidade de escolher determinadas subcaracterísticas da qualidade para efetuar uma dada avaliação, sem ter que necessariamente avaliar toda uma característica.

## 1.2 RELEVÂNCIA DO TRABALHO

Com o aumento da demanda e competitividade entre os softwares, cada vez mais as empresas estão em busca da qualidade para seus produtos. De encontro a esta necessidade, as normas internacionais ISO/IEC 9126 e NBR ISO/IEC 12119, apresentam requisitos para

avaliação de softwares, proporcionando maior abertura de mercado para os produtos que seguirem suas especificações.

Em função desta importância, torna-se essencial a avaliação destes produtos em relação à qualidade. A grande dificuldade da avaliação de um software através das Normas ISO/IEC 9126 e NBR ISO/IEC 12119 é que a avaliação pode gerar resultados bastante diferentes, dependendo da experiência e do entendimento dos avaliadores. Assim, a complexidade do trabalho reside, justamente, na redução da subjetividade das normas citadas. Com isso, deve-se elaborar um *checklist* bastante objetivo, de forma que os resultados obtidos por diferentes avaliadores sejam semelhantes. Além disso, o avaliador poderá incluir novos itens ao *checklist* já proposto pelo software, bem como estabelecer pesos diferenciados para cada característica do *checklist*.

### 1.3 ESTRUTURA DO TRABALHO

**[F1] Comentário:** Inserir o item 1.2 para falar sobre a relevância / justificativa / delimitações do trabalho.

O primeiro capítulo apresenta uma introdução ao trabalho, iniciando alguns conceitos empregados em sua elaboração. São apresentados ainda, os objetivos e a organização do texto.

O segundo capítulo descreve o conceito do termo Qualidade sob a influência da globalização contendo algumas considerações. Aborda também a qualidade nas organizações, a questão da qualidade voltada à área de software, apresentando as definições existentes para a qualidade de software.

O terceiro e o quarto capítulo compilam a fundamentação teórica reunindo os conceitos e fundamentações das normas técnicas ISO/IEC 9126 e NBR ISO/IEC 12119, respectivamente.

O quinto capítulo apresenta a especificação da ferramenta onde consta o diagrama de contexto, diagrama fluxo de dados, diagrama de entidade e relacionamento (DER) e o dicionário de dados. Também no quinto capítulo é apresentada a ferramenta para avaliação da qualidade de produtos de software, incluindo seu funcionamento e exemplos de telas.

No sexto capítulo são feitas as considerações finais sobre o trabalho, incluindo conclusões e sugestões.

## 2 QUALIDADE

### 2.1 CONCEITO DE QUALIDADE

Segundo Campos (1994), na era de economia global não é mais possível garantir a sobrevivência da empresa apenas exigindo que as pessoas façam o melhor que puderem ou cobrando apenas resultados. São necessários métodos que possam ser utilizados por todos em direção aos objetivos de sobrevivência da empresa. Estes objetivos podem ser traduzidos em uma única frase: “satisfação das necessidades de todas as pessoas”.

A economia está caminhando para uma nova era, afirma Inthurn (2001), a da globalização. Todos os dias as empresas confrontam-se mais e mais com o impacto causado pelo fenômeno da globalização. Neste contexto, se deparam com o elemento mais impactante que é a informática. Esta revolução tecnológica nos permite hoje, transmitir dados, informações e conhecimento para todos os continentes.

As empresas que desenvolvem software passaram a valorizar não somente o seu produto, mas todas as fases que envolvem a sua elaboração e posterior entrega. Assim, a preocupação passou a ser com seu capital intelectual e o comprometimento com a qualidade, que é fator de competitividade e se mostra cada vez mais presente no plano estratégico das organizações. Este novo cenário sócio-econômico mundial, voltado à qualidade, tem forçado as organizações a reverem sua postura frente ao consumidor, ao empregado e à sociedade em geral.

A qualidade tem sido a responsável pelo aumento no grau de competitividade de inúmeras empresas. No entanto, apesar das vantagens associadas à qualidade, existe um grande caminho a se percorrer, no sentido de conscientizar as organizações sobre estas vantagens e sobre as ações que devem ser executadas para alcançar a qualidade (Storch, 2000).

Havendo liberdade de escolha, os clientes das organizações públicas e privadas fazem uma serie de exigências. Tudo aquilo que estão acostumados a obter torna-se, para eles, qualidade obrigatória. Para conquistar clientes, diante da competição atual, é preciso encantá-los e seduzi-los. Qualidade intrínseca, preço acessível, flexibilidade, baixo custo de manutenção, valor de revenda, prazo de entrega, rapidez, variedade de opções, cordialidade



no atendimento, boas condições de pagamento, imagem no mercado, segurança pessoal e ambiental são algumas entre as muitas possibilidades exploradas como vantagens competitivas.

Diante das definições citadas acima, pode-se concluir que a qualidade apresenta-se então como uma necessidade e não como um mero modismo. Qualidade pode ser tudo aquilo que alguém faz ao longo de um processo para garantir que um cliente, interno ou externo à organização, obtenha exatamente aquilo que deseja.

## 2.2 QUALIDADE SOFTWARE

A qualidade de software não deve ser encarada como apenas um diferencial de mercado para que uma dada empresa de software possa conseguir vender e lucrar, mas sim como um pré-requisito que a empresa deve conquistar para conseguir colocar o produto no Mercado Global (Sanders, 1994).

Segundo a norma ISO/IEC 9126 (2000), “*qualidade é a totalidade das características de uma entidade que lhe confere a capacidade de satisfazer as necessidades explícitas e implícitas*”. Esta definição formal merece alguns complementos, principalmente para definir o que são as entidades, as necessidades explícitas e as necessidades implícitas. A entidade é o produto do qual se está falando, que pode ser um bem ou um serviço. As necessidades explícitas são as próprias condições e objetivos propostos pelo produtor. As necessidades implícitas incluem as diferenças entre os usuários, a evolução no tempo, as implicações éticas, as questões de segurança e outras visões subjetivas.

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 o [Quadro 2-1](#).

Excluído: Quadro 2-1

Excluído: Quadro 2-1

**Quadro 2-1 - Fatores explícitos**

Fatores	Descrição
Prazo do projeto	Prazo desejado ou estabelecido pelo cliente
Progresso	Informações sobre o progresso do projeto disponibilizado
Atendimento funcional	Extensão em que o sistema satisfaz suas especificações
Confiabilidade	Extensão em que o sistema desempenha suas funções requeridas
Integridade	O acesso às informações pode ser controlado
Usabilidade	Esforço requerido para aprender a operar
Investimentos	Benefícios econômicos obtidos pelo cliente através do sistema
Atendimento	Tempos de atendimento para manutenções no produto

Fonte: (Storch, 2000, p.23)

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 o [Quadro 2-2](#),

**Quadro 2-2 – Fatores implícitos**

Fatores	Descrição
Manutenibilidade	Esforço requerido para localizar e remover um defeito de um módulo
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: (Storch, 2000, p.23)

Excluído: Quadro 2-3

Excluído: Quadro 2-4

Excluído: 3

Excluído: 4

## 2.2.1 IMPORTÂNCIA DA QUALIDADE DE SOFTWARE

A qualidade, hoje em dia, é essencial para a sobrevivência e o sucesso de empresas de software. Uma organização não sobressairá tanto no mercado nacional quanto no mercado global a menos que produza software de boa qualidade e seus clientes percebam seus produtos e serviços como sendo de boa qualidade (Sanders, 1994).

Para isso, existem muitas razões que devem ser levadas em conta, a saber:

- a) qualidade é competitividade: a única maneira de diferenciar o produto do competidor é pela qualidade do software e do suporte que é fornecido juntamente. Como o mercado amadurece, usuários não querem apenas que a empresa fale que tem qualidade, mas que mostre a todos a sua qualidade através de Certificação internacional. Não ter certificação pode acarretar desvantagem competitiva;
- b) qualidade é essencial para a sobrevivência: clientes estão pedindo por qualidade. Se a empresa não tiver habilidade de sobreviver em um mercado altamente competitivo, ela está em débito com o mercado. A maioria das grandes organizações está reduzindo o número de fornecedores, e um meio de escolher os fornecedores é verificando quais deles têm certificações de qualidade;
- c) qualidade é essencial para o mercado internacional: O mercado de software está, cada vez mais, se tornando global. A habilidade das empresas de mostrarem qualidade, eventualmente as colocam no mercado global. O mercado local é vulnerável a produtos importados que, normalmente, têm mais qualidade;
- d) qualidade é custo/benefício: um sistema de qualidade direciona para o aumento da produtividade e permanentemente reduz custos, habilitando o gerenciamento para reduzir a correção de defeitos dando ênfase à prevenção. Todas as empresas sabem que corrigir defeitos após o desenvolvimento do software é mais dispendioso do que corrigi-los depois. Prevenir defeitos primeiramente pode resolver muita coisa depois e economizar bastante;
- e) qualidade retém consumidores e aumenta lucros: pouca qualidade normalmente custa muito mais do que contratar mais desenvolvedores e ainda continuar sem qualidade. A maioria dos consumidores não tolerará a falta de qualidade e irão procurar outros desenvolvedores. Mais qualidade aumenta a satisfação dos consumidores e assegura os que já são clientes há mais tempo;

## 2.2.2 O QUE É UM SISTEMA DA QUALIDADE?

Aplicar os princípios da qualidade de software é o início para o sucesso. O termo “sistema da qualidade” é utilizado internacionalmente para descrever um processo na qual garante e demonstra a qualidade dos produtos e serviços ofertados pela empresa.

A norma ISO Série 9000, define e descreve o que é requerido ou satisfatório em um sistema de qualidade contendo componentes de desenho e desenvolvimento (as padronizações existentes para garantir a qualidade de software serão detalhadas nos capítulos 3 e 4).

Além das padronizações ISO, muitas outras organizações nacionais e internacionais promovem padrões que descrevem sistemas de qualidade para serem aplicados em sistemas de desenvolvimento e suporte em certas circunstâncias, como é o caso do *Capability Maturity Model – CMM*.

O termo “gerenciamento de sistemas da qualidade” é utilizado algumas vezes, enfatizando as necessidades dos processos da qualidade para serem gerenciados para garantir que continue de forma correta e eficiente.

Tão importante quanto as práticas e ferramentas é o status da pessoa que as usa. A qualidade deve garantir que as pessoas envolvidas devem ter suas habilidades certas para cada tipo de trabalho de uma maneira profissional. Se as pessoas necessitam de treinamento, então a empresa deverá treinar os seus usuários. Deve-se garantir que as pessoas entendam suas responsabilidades e como seu trabalho se relaciona com outras pessoas.

Um sistema da qualidade dá grande ênfase à correção de erros. É muito útil corrigir os erros durante o início do ciclo de vida do sistema. Melhor ainda, é anular erros antes mesmo deles serem feitos. Um sistema de qualidade e sucesso inclui maneiras de registrar os erros para determinar as causas e agir de acordo com o erro eliminando suas causas.

Em suma, um sistema da qualidade é tudo que o gerenciamento utiliza para garantir e demonstrar a qualidade do software e do serviço de suporte. O sistema da qualidade é o trabalho completo, incluindo política, procedimentos, ferramentas e recursos, incluindo humano e tecnológico.

## **2.2.3 VISÕES DA QUALIDADE DE SOFTWARE**

Segundo Storch (2000), não existe um sistema de classificação de software amplamente aceito, o que existe são sistemas de classificação de software específicos para cada classe de software, ou seja, a importância de cada característica da qualidade é atribuída 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, portanto, será verificado cada ponto de vista em separado, compreendendo a visão do usuário, a visão da equipe de desenvolvimento e a visão do gerente.

### **2.2.3.1 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 podem 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.

### **2.2.3.2 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.

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

### 2.2.3.3 VISÃO DO GERENTE

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

## 2.3 QUALIDADE DE PRODUTOS DE SOFTWARE

Segundo Rocha (2001), qualidade de software pode ser vista como um conjunto de características que devem ser alcançadas em um determinado grau para que o produto atenda às necessidades de seus usuários. É por meio desse conjunto de características que a qualidade de um produto de software pode ser descrita e avaliada.

Cada uma das características pode ser detalhada em vários níveis de subcaracterísticas, chegando-se a um amplo conjunto de atributos que descrevem a qualidade de um produto de software. Torna-se, então, necessário escolher um modelo que organize os atributos e permita avaliar a qualidade de software. Esses modelos nos permitem, ainda, entender como as diferentes facetas contribuem para a qualidade do produto como um todo. A norma ISO/IEC 9126 surgiu como uma importante tentativa de consolidar as diferentes visões da qualidade em um modelo como norma internacional.

Boegh (1999), descreve um método (*Squid – Software Quality in Development*) pelo qual uma organização desenvolvedora de software pode usar medições para planejar e controlar a qualidade do produto durante o desenvolvimento, avaliar a qualidade do produto final e aprender sobre o processo de software e sobre o produto. Dessa maneira obtém-se um *feedback* para melhorar o projeto em andamento e os novos projetos em desenvolvimento.

Outro aspecto importante está relacionado à qualidade dos pacotes de software (seleção e aquisição). Para selecionar e adquirir um pacote de software deve-se considerar as necessidades específicas que se quer atender e a qualidade dos produtos de software disponíveis no mercado que atendam a essas necessidades. A NBR ISO/IEC 12119 estabelece os requisitos de qualidade de um pacote de software e fornece instruções sobre como testar o software com relação a esses requisitos.

A norma ISO/IEC 9126 apresenta um conjunto de características da qualidade aplicável a qualquer produto de software. Entretanto, produtos em domínios de aplicação específicos e as diferentes tecnologias utilizadas no desenvolvimento de produtos de software implicam características específicas que determinam a qualidade desses produtos. Tais características precisam ser identificadas de modo a obter a qualidade desejada.

O objetivo, ao se desenvolver um produto de software, não é alcançar a qualidade perfeita, mas sim a qualidade necessária e suficiente para o uso especificado, quando o produto for entregue e realmente usado pelos usuários (Collins, 1994). Qualidade custa caro e não é possível atingir todas as características de qualidade no mais alto grau (Boegh, 1993).

Torna-se, portanto, necessário identificar quais são as características de qualidade necessárias para um determinado produto de software e definir em que grau essas características precisam ser alcançadas para satisfazer as necessidades dos usuários.

## **2.4 NORMAS DE QUALIDADE DOS PRODUTOS DE SOFTWARE**

Entidades de normalização de mais de noventa países fazem parte do grupo *International Organization for Standardization* (ISO - Organização Internacional de Normalização), entre os quais o Brasil, através da *Associação Brasileira de Normas Técnicas* (ABNT) (Antonioni, 1995).

Para realizar o acompanhamento da qualidade são estabelecidos procedimentos, parâmetros e medidas para verificar se um produto está dentro de limites aceitáveis. No caso do produto de software, essas medidas são difíceis de serem definidas, pois ele possui características que o diferenciam dos outros produtos. Não existem medidas que possam afirmar que um produto de software está totalmente correto ou não. No máximo pode-se dizer que ele funcionou em todas as circunstâncias em que foi testado (Antonioni, 1995).

Surgiu então, a necessidade de criar normas para estabelecer uma forma de avaliação da qualidade de produtos de software a partir de características que devem ser observadas para que ele seja considerado um produto de qualidade.

No **Quadro 2-3**, são apresentadas as principais normas nacionais e internacionais existentes na área de software.

Excluído: Quadro 2-5

Excluído: Quadro 2-7

**Quadro 2-3 – Normas nacionais e internacionais na área de software**

Excluído: 5

Excluído: 7

Norma	Comentário
ISO 9126	Características da qualidade de produtos de software.
NBR 13596	Versão brasileira da ISO 9126.
ISO 14598	Guias para a avaliação de produtos de software, baseados na utilização prática da Norma ISO 9126.
ISO 12119	Características de qualidade de pacotes de software
IEEE P1061	<i>Standard for Software Quality Metrics Methodology</i> (Normas para Métricas de Qualidade de Software).
ISO 12207	<i>Software Life Cycle Process</i> (Norma para a qualidade do processo de desenvolvimento de software).
NBR ISO 9001	Sistemas de qualidade – modelo para garantia de qualidade em projeto, desenvolvimento, instalação e assistência técnica (processo).
NBR ISO 9002	Modelo para garantia da qualidade em produção, instalação e serviços
NBR ISO 9003	Gestão de qualidade e garantia da qualidade. Aplicação da norma ISO 9000 para o processo de desenvolvimento de software.
NBR ISO 10011	Auditoria de Sistemas de Qualidade (processo).

Fonte: (Barreto, 2002)

O Subcomitê de Software (SC7) do Comitê Técnico Conjunto (JTC1) da ISO e IEC vêm trabalhando na elaboração de normas e relatórios técnicos que permitam especificar e avaliar a qualidade dos produtos de software. Essas normas e relatórios são definidos pelos seguintes documentos:

- a) qualidade de produto de software**
  - i) ISO/IEC 9126-1: Modelo de Qualidade
  - ii) ISO/IEC 9126-2: Métricas Externas
  - iii) ISO/IEC 9126-3: Métricas Internas



- iv) ISO/IEC 9126-4: Métricas da Qualidade em Uso
- b) avaliação de produtos de software**
  - i) ISO/IEC 14598-1: Visão Geral
  - ii) ISO/IEC 14598-2: Planejamento e Gestão
  - iii) ISO/IEC 14598-3: Processo para Desenvolvedores
  - iv) ISO/IEC 14598-4: Processo para Adquirentes
  - v) ISO/IEC 14598-5: Processo pra Avaliadores
  - vi) ISO/IEC 14598-6: Documentação de Módulos de Avaliação
- c) teste e requisitos de qualidade em pacotes de software**
  - i) NBR ISO/IEC 12119: Pacotes de Software – Teste e requisitos de qualidade

## **3 NORMAS ISO/IEC 9126 E NBR ISO/IEC 12119**

### **3.1 NORMA ISO/IEC 9126 (NBR 13596)**

Segundo Inthurn (2001), a qualidade de produtos de software, na maioria dos casos, não é dada pelo produtor do software ao usuário final. Ocorrem repetidas vezes que o próprio usuário não sabe como exigir esta qualidade ou mesmo não tem a intenção de exigí-la. Muitas vezes isto ocorre porque alguns ainda julgam difícil avaliar a qualidade de um software.

Quando se pensa em qualidade de um produto, é fácil imaginar padrões de comparação, provavelmente ligados às dimensões do produto (comprimento, largura, etc) ou alguma outra característica (cor, peso, textura, etc). Porém para software é diferente, pois se lida com produtos não mensuráveis fisicamente, mas com produtos lógicos que requerem padrões de verificação de qualidade diferenciados do habitual.

A *International Organization for Standardization* (ISO) publicou uma norma que representa a atual padronização mundial para a qualidade de produtos de software. Esta norma chama-se ISO/IEC 9126 e foi publicada em 1991. Ela é uma das mais antigas da área de qualidade de software e possui sua tradução para o Brasil, publicada em agosto de 1996 como NBR 13596.

A norma ISO/IEC 9126, em conjunto com a norma ISO/IEC 14598, descreve um modelo de qualidade que pode ser utilizado por organizações que pretendam fazer avaliações de produtos de software. O processo de avaliação de software utilizado neste trabalho segue este modelo, o qual será tratado no item a seguir. Este guia mostra em detalhes como poderão ser aplicadas as normas numa avaliação da qualidade. Devido a importância deste modelo e cada um de seus processos, o mesmo será evidenciado no item 3.1.

#### **3.1.1 PROCESSO DE AVALIAÇÃO DE PRODUTOS DE SOFTWARE**

O guia para utilização de normas sobre qualidade de produtos de software (ISO/IEC 9126 e 14598) tem como objetivo facilitar a compreensão dessas normas, demonstrando uma

visão completa sobre o processo de avaliação de produtos de software. Este guia foi elaborado considerando o seguinte público:

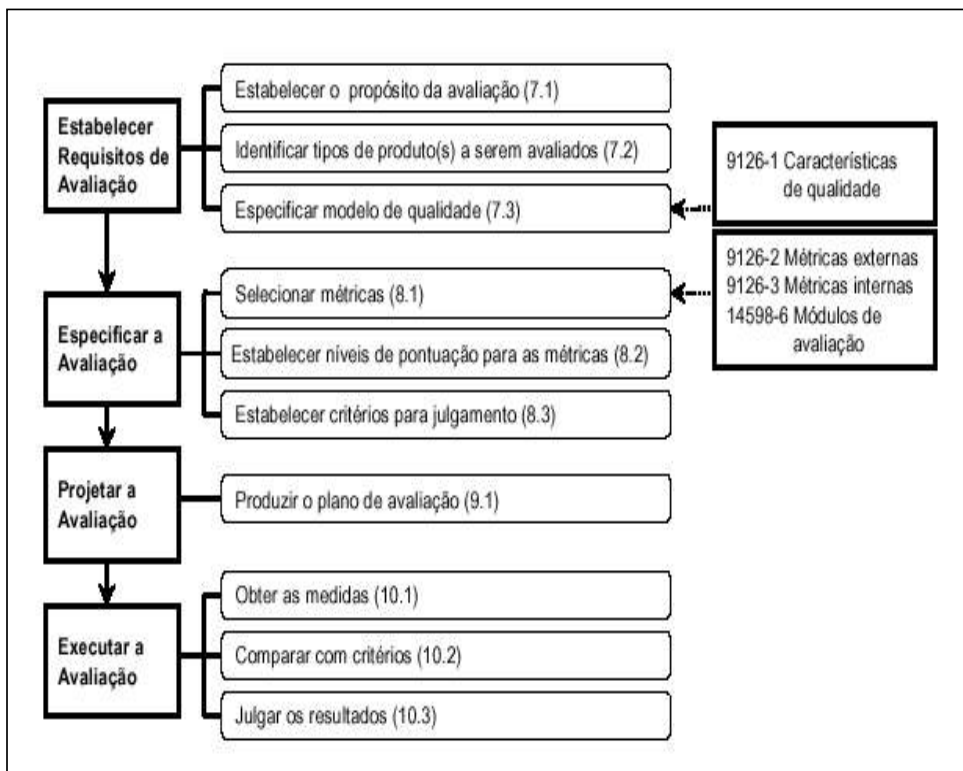
- Desenvolvedores de software que pretendam utilizar a avaliação de produtos de software como forma de aprimorar seu processo de desenvolvimento e conseqüentemente melhorar a qualidade do produto de software;
- Pessoas ou organizações que adquirem software e que pretendem utilizar a avaliação como um processo de seleção desses produtos;
- Organizações que executem avaliação independente de produto de software e que pretendam utilizar normas internacionais como referência para a avaliação;
- Professores e estudantes com interesse em qualidade de software.

A [Figura 3-1](#) mostra em detalhes as etapas do processo de avaliação de produtos de software. Cada uma destas etapas será apresentada em maiores detalhes nos itens a seguir.

Excluído: Figura 3-1

Excluído: Figura 3-1

**Figura 3-1 - Processo de avaliação**



Fonte: (Koscianski, 1999)

### **3.1.1.1 ESTABELECEER REQUISITOS DE AVALIAÇÃO**

Esta etapa está dividida em três atividades: estabelecer o propósito da avaliação, identificar tipos de produtos a serem avaliados e especificar modelo de qualidade.

A avaliação de produto de software baseia-se na comparação do produto contra alguns requisitos, ou ainda contra necessidades explícitas e implícitas dos usuários. Porém, o objetivo específico da avaliação depende do ponto de vista do usuário, seja ele um desenvolvedor (usuário sob o ponto de vista de gente de manutenção de software) ou adquirente do produto de software.

O tipo de produto a ser avaliado depende do propósito da avaliação. Como um primeiro passo, convém que o avaliador defina os produtos a serem avaliados como intermediários (durante o ciclo de vida de desenvolvimento) ou produtos finais.

O modelo de qualidade especificado para a avaliação serve como referência para a definição dos requisitos de qualidade para o produto de software. Neste estágio da avaliação, os requisitos são descritos para as características de qualidade mais relevantes, sendo priorizados de acordo com as necessidades dos usuários.

### **3.1.1.2 ESPECIFICAR A AVALIAÇÃO**

Esta etapa está dividida em três atividades: selecionar métricas, estabelecer níveis de pontuação para as métricas e estabelecer critérios para julgamento.

A especificação e medição quantitativa dos requisitos de qualidade do produto de software só podem ser feitas pelo uso de métricas associadas às características de qualidade esperadas. Métricas podem ser:

- a) internas: associadas à arquitetura do produto de software e que permitem prever a qualidade do produto final;
- b) externas: mensuráveis quando o produto está em operação;
- c) de qualidade em uso: que avaliam o efeito do uso do produto de software.

Para cada métrica selecionada devem ser definidos os níveis de pontuação e a escala relacionada, onde poderá ser representado o nível requerido para o atributo a ser medido. Para julgar a qualidade do produto, convém que o avaliador prepare um

procedimento para este fim, com critérios específicos para as diferentes características de qualidade, seja em termos de cada subcaracterísticas, ou através de uma combinação ponderada das subcaracterísticas. Outros aspectos a serem considerados no julgamento de um produto de software num ambiente específico são prazo e custo.

### 3.1.1.3 PROJETAR A AVALIAÇÃO

Esta etapa define as atividades do plano de avaliação, descrevendo os métodos de avaliação e o cronograma de atividades do avaliador.

### 3.1.1.4 EXECUTAR A AVALIAÇÃO

Esta etapa está dividida em três atividades: obter as medidas, comparar com critérios e julgar os resultados.

As métricas selecionadas são aplicadas ao produto de software, resultando em valores nas escalas das métricas. Os valores medidos são então comparados aos critérios estabelecidos na especificação da avaliação. Na atividade de julgamento, um conjunto de valores pontuados é sintetizado e é feita uma declaração sobre quanto o produto de software atende aos requisitos de qualidade. Esta síntese é também comparada com outros aspectos como prazo e custo. Finalmente, através de critérios gerenciais, pode ser tomada uma decisão quanto à aceitação ou rejeição do produto de software, ou quanto à sua liberação ou não para uso. Os resultados da avaliação influenciam os próximos passos do ciclo de desenvolvimento do software, através de decisões como “convém que os requisitos sejam modificados, ou são necessários mais recursos para o processo de desenvolvimento?”. (Weber, 2001).

Na [Figura 3-2](#), é demonstrado um exemplo de escalas para uma determinada métrica de avaliação de produtos de software.

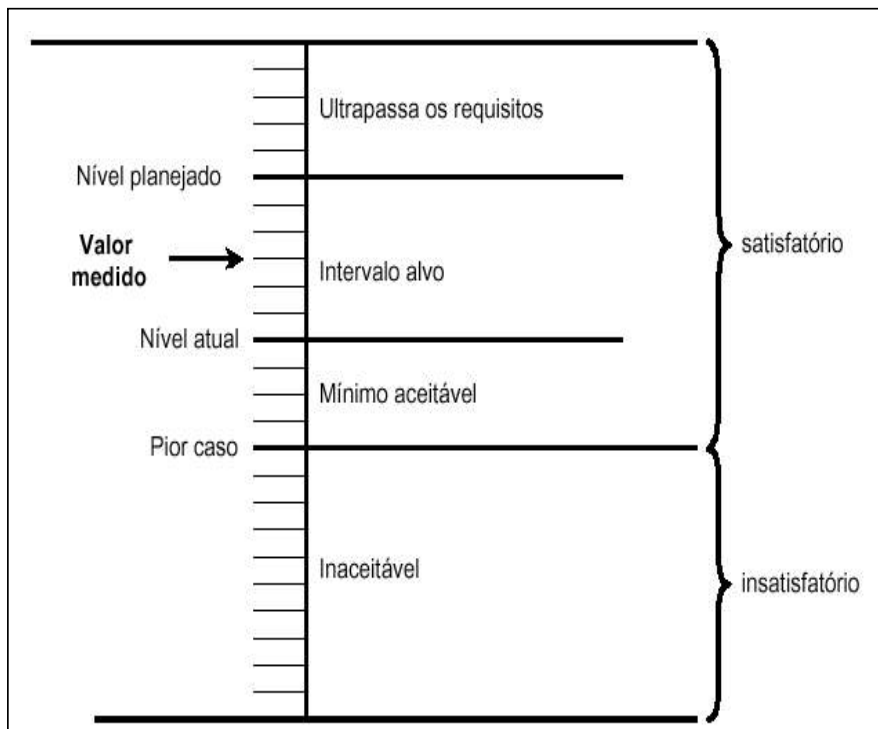
Excluído: Figura 3-3

Excluído: Figura 3-4

Figura 3-2 - Níveis de pontuação para as métricas

Excluído: 3

Excluído: 4



Fonte: (Koscianski, 1999)

### 3.1.2 MODELO DA QUALIDADE DEFINIDO PELA NORMA ISO/IEC 9126

De acordo com a norma ISO/IEC 9126 (ISO/IEC 9126, 2001), a avaliação de um produto de software deveria ser:

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

O modelo da qualidade definido na ISO/IEC 9126 (ISO/IEC 9126, 2001), utilizado como referência para o processo de avaliação da qualidade de produto de software, está

subdividido em duas partes: modelo para características externas e internas; e modelos para qualidade em uso.

### **3.1.2.1 MODELO PARA CARACTERÍSTICAS EXTERNAS E INTERNAS**

O modelo da qualidade para características externas e internas classifica os atributos de qualidade dos produtos de software em seis características: funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade. Estas características, por sua vez, são desdobradas em subcaracterísticas. As subcaracterísticas podem ser desdobradas em mais níveis, que caracterizam os atributos da qualidade. As métricas internas e externas aplicam-se, em geral, ao nível dos atributos da qualidade.

#### **3.1.2.1.1 FUNCIONALIDADE**

Refere-se à existência de um conjunto de funções que satisfazem necessidades explícitas ou implícitas e suas propriedades específicas. Tem como subcaracterísticas: adequação, acurácia, interoperabilidade, conformidade e segurança de acesso.

- a) adequação: atributos de software que influenciam na presença de adequação de um conjunto de funções para tarefas específicas e objetivos do uso;
- b) acurácia: atributos do software que evidenciam a presença de resultados ou efeitos corretos ou conformidade acordados;
- c) interoperabilidade: atributos de software que influenciam na habilidade de interagir com um ou mais sistemas específicos;
- d) conformidade: atributos do software que influenciam na aderência e padrões relativos a convenções ou regulamentações legais e prescrições similares;
- e) segurança de acesso: atributos de software que influenciam na habilidade de prevenir acessos não intencionados e resistir a ataques intencionados para se ter acesso não autorizado à informação confidencial, ou fazer modificações não autorizadas em informações ou em programa.

#### **3.1.2.1.2 CONFIABILIDADE**

Refere-se à capacidade do software manter seu nível de desempenho, sob condições estabelecidas, por um período de tempo. Tem como subcaracterísticas: maturidade, tolerância a falhas, recuperabilidade e conformidade.

- a) maturidade: atributos de software que influenciam na frequência de erros devido a falhas no software;
- b) tolerância a 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) 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;
- d) conformidade: atributos de software que influenciam na capacidade do software manter seu nível de desempenho conforme padrões estabelecidos.

### **3.1.2.1.3 USABILIDADE**

Refere-se ao esforço necessário ao uso do produto de software, bem como o julgamento individual de tal uso, por um conjunto explícito ou implícito de usuários. Tem como subcaracterísticas: inteligibilidade, apreensibilidade, operacionalidade, atratividade e conformidade.

- a) 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) apreensibilidade: atributos de software que influenciam a facilidade com a qual o usuário pode aprender suas aplicações;
- c) operacionalidade: atributos de software que influenciam no esforço necessário para o usuário poder operar e manter o controle da operação.
- d) atratividade: atributos de software que influenciam o interesse do usuário pelo uso do software;
- e) conformidade: atributos de software que influenciam no esforço necessário para se poder utilizar o software da maneira que foi especificado.

### **3.1.2.1.4 EFICIÊNCIA**

Refere-se ao relacionamento entre o nível de desempenho do software e a quantidade de recursos utilizada, sob condições estabelecidas. Tem como subcaracterísticas: comportamento em relação ao tempo, comportamento em relação aos recursos e conformidade.



- a) comportamento em relação ao tempo: atributos do software que influenciam no tempo de resposta e processamento e desempenho na execução de suas funções;
- b) comportamento em relação aos recursos: atributos de software que influenciam na quantidade de recursos usados e a duração de tal uso na execução de suas funções.
- c) conformidade: atributos de software que influenciam na relação entre nível de desempenho do software e a quantidade de recursos utilizada, sob condições estabelecidas.

#### **3.1.2.1.5 MANUTENIBILIDADE**

Refere-se ao esforço necessário para fazer modificações específicas no software. Tem como subcaracterísticas: analisabilidade, modificabilidade, estabilidade, testabilidade e conformidade:

- a) analisabilidade: atributos de software que influenciam na necessidade de recursos para diagnóstico de deficiências ou causas de falhas, ou para identificação de partes a serem modificadas;
- b) modificabilidade: atributos de software que influenciam na necessidade de recursos para implementar as modificações específicas;
- c) estabilidade: atributos de software que influenciam nos riscos de efeitos inesperados das modificações;
- d) testabilidade: atributos de software que influenciam na necessidade de recursos necessários para validação do software modificado;
- e) conformidade: atributos de software que influenciam nas modificações específicas do software tal como foram necessitadas.

#### **3.1.2.1.6 PORTABILIDADE**

Refere-se à habilidade do software para ser transferido de um ambiente para outro. Tem como subcaracterísticas: adaptabilidade, capacidade de instalação, conformidade, capacidade de substituição e coexistência.

- a) 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) capacidade de instalação: atributos de software que influenciam nos esforços necessários para instalar o software no ambiente especificado;

- c) conformidade: atributos de software que fazem o software manter padrões ou convenções relativas comuns;
- d) 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.
- e) coexistência: atributos de software que requerem esforços para existir no mesmo ambiente que outros softwares sem sofrer influências dos mesmos ou influenciá-los.

A ISO/IEC 9126-1 conduz a um entendimento dos conceitos que definem as diversas características e subcaracterísticas da qualidade de produto de software; porém, na prática, ainda não facilita o suficiente a definição dos requisitos de qualidade a partir dela. As definições de características de qualidade permitem perceber um possível universo de requisitos que se enquadram no conceito apresentado, mas dificilmente permitiriam elaborar uma declaração de requisitos mais criteriosa, quanto às mesmas. Não faz sentido, por exemplo, uma declaração do tipo “o produto de software deve ter uma usabilidade de 0,5”, pois esse número não teria qualquer significado (pelo menos no estado-da-arte em que se encontra o tema de medição da qualidade de produto de software) (Weber, 2001).

O primeiro desdobramento em subcaracterísticas serve para delimitar melhor o amplo universo contemplado pela característica. Introduce conceitos mais detalhados que facilitam a especificação de requisitos, ajudando a pensar na característica da qualidade a partir de seus componentes. Mas este desdobramento ainda não é suficiente para especificar os requisitos da qualidade. Uma declaração do tipo “A operacionalidade deve ser igual a 0,8”, por exemplo, continua sem fazer sentido.

Assim o usuário da norma que necessite elaborar sua declaração de requisitos deve fazer o próximo nível de desdobramento, os *atributos*, (que não estão presentes na ISO/IEC 9126), identificando aspectos relevantes ao produto de software e que se enquadrem nas características e subcaracterísticas citadas. Desta forma, uma declaração do tipo “O tempo de uso do produto de software até que se tenha domínio operacional do mesmo deverá ser inferior a 20 horas”, por exemplo, é adequada como requisito da subcaracterística operacionalidade, que faz parte da característica usabilidade.

### 3.1.2.2 MODELO PARA QUALIDADE EM USO

Os atributos da qualidade em uso são classificados em quatro características: efetividade, produtividade, segurança e satisfação. A qualidade em uso é “a capacidade do produto de software de permitir a usuários específicos atingir metas especificadas com efetividade, produtividade, segurança e satisfação em um contexto de uso especificado” (Weber, 2001).

- a) efetividade: refere-se à capacidade do produto de software em possibilitar aos usuários atingir metas especificadas com acurácia e completeza, num contexto de uso especificado;
- b) produtividade: refere-se à capacidade do produto de software em possibilitar aos usuários utilizar uma quantidade adequada de recursos em relação à efetividade alcançada, num contexto de uso especificado;
- c) segurança: refere-se à capacidade do produto de software em propiciar níveis aceitáveis de risco de danos a pessoas, negócios, software, propriedade ou ambiente, num contexto de uso especificado;
- d) satisfação: refere-se à capacidade do produto de software em satisfazer usuários, num contexto de uso especificado.

A visão do modelo da qualidade, levando em conta os resultados obtidos a partir do uso do software no ambiente especificado, é uma inovação em relação ao modelo original da ISO/IEC 9126:1991. Esta visão pode ser usada como referência para a definição de requisitos da qualidade esperados para o ambiente de uso, assim como para a avaliação dos resultados obtidos. Quando o modelo de qualidade em uso for utilizado para a definição de requisitos, é necessário o estabelecimento das regras de derivação de atributos de características externas para o produto de software a partir das características da qualidade em uso.

## 3.2 NORMA NBR ISO/IEC 12119

Apesar da grande virtude da Norma ISO/IEC 9126 em definir as características da qualidade de software de modo geral, há uma grande dificuldade na sua aplicação prática devido às características e particularidades dos produtos de software.

Com base nas experiências de avaliação da qualidade dos produtos de software já publicadas, percebe-se a necessidade de um maior detalhamento ou adequação da Norma ISO/IEC 9126 para cada grupo de produtos de software <sup>1</sup> que se pretende avaliar. A Norma NBR ISO/IEC 12119 vem justamente definir com profundidade cada uma das características e subcaracterísticas da Norma ISO/IEC 9126, aplicável a pacotes de software (Sandri, 1998).

Conforme Rocha (2001), a NBR ISO/IEC 12119 tem como objetivo estabelecer quais os requisitos de qualidade de um software tipo pacote e fornece instruções para testar esse software em relação aos requisitos definidos. Seu escopo refere-se a pacotes de software na forma oferecida no mercado, e não aos processos de desenvolvimento de software.

Segundo Sandri (1998), pacote de software é o conjunto completo e documentado de programas fornecidos a diversos usuários para uma aplicação ou função genérica tais como: processadores de texto, planilhas eletrônicas, banco de dados, software gráficos, programas para funções técnicas ou científicas e programas utilitários.

A NBR ISO/IEC 12119 pode ser utilizada pelo produtor do software para realizar uma comparação com produtos semelhantes no mercado e auxiliar no desenvolvimento, através de um processo de evolução contínua de avaliação do produto. Esta norma pode ainda ser utilizada por potenciais usuários que desejam saber se um determinado pacote de software irá satisfazer as suas necessidades; bem como pode ser utilizada por entidades de certificação, laboratórios de teste, auditores dentre outros.

De acordo com a Associação Brasileira de Normas Técnicas (1998), a NBR ISO/IEC 12119 define os requisitos da qualidade para pacotes de software e instruções para testes em relação a estes requisitos. Na **Figura 3-3** **Erro! Fonte de referência não encontrada.**, está

<sup>1</sup> Entende-se por grupo de produtos de software os produtos que possuem os mesmos objetivos funcionais ou as mesmas características.

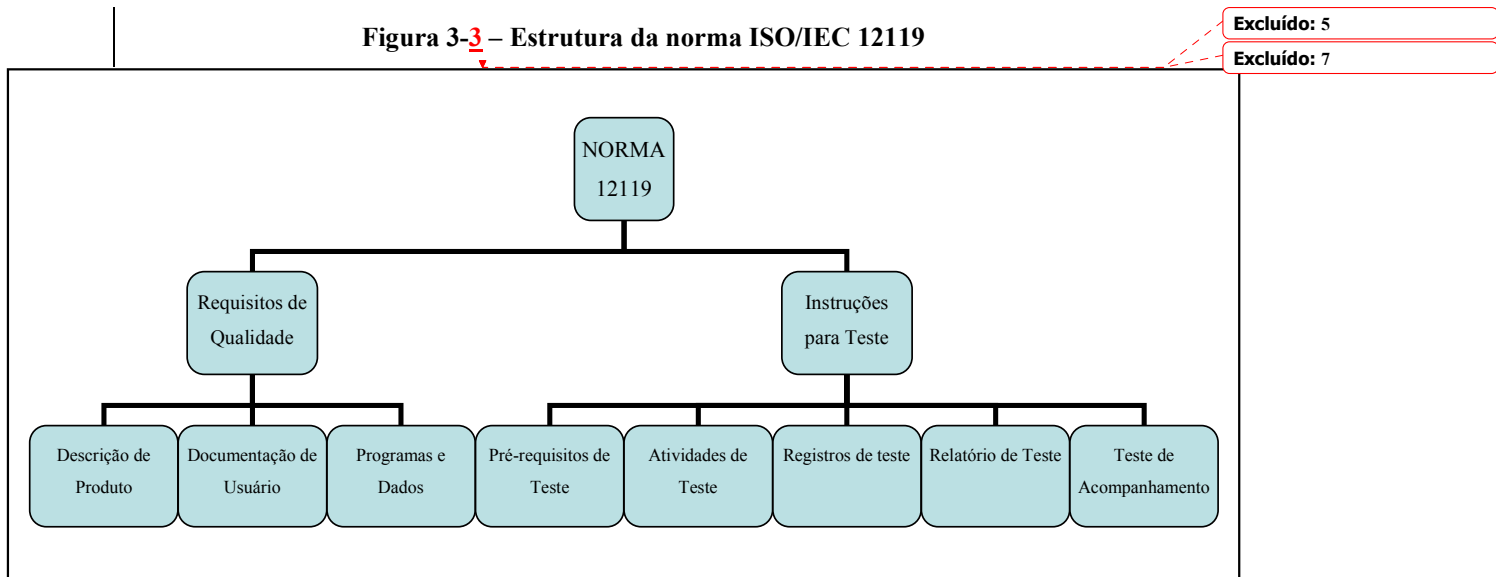
**Excluído:** Figura 3-5

**Excluído:** Figura 3-7

**Excluído:** Erro! A origem da referência não foi encontrada.

estruturada a norma ISO/IEC 12119 onde é possível visualizar melhor as divisões e subdivisões desta norma.

**Figura 3-3 – Estrutura da norma ISO/IEC 12119**



Fonte: (Sandri, 1998)

Os requisitos da qualidade para pacotes de software são classificados em: descrição de produto, documentação de usuário e programas e dados.

As instruções para teste são classificadas em: pré-requisitos de teste, atividades de teste, registros de teste, relatório de teste e teste de acompanhamento.

Nas subseções que se seguem, será descrito cada um dos requisitos de qualidade e instruções para teste, mas antes é importante esclarecer que, para o entendimento dos requisitos de qualidade, se faz necessário conhecer a definição de cada uma das seis características da qualidade para produtos de software (funcionalidade, confiabilidade, usabilidade, eficiência, manutenibilidade e portabilidade), as quais foram descritas no capítulo 3.

Um pacote de software está em conformidade com a Norma NBR ISO/IEC 12119 se atende a todos os requisitos definidos nas subseções que se seguirão. As recomendações indicadas pelo uso da forma verbal “convêm que” são opcionais.

### **3.2.1 REQUISITOS DA QUALIDADE PARA DESCRIÇÃO DE PRODUTO**

Segundo a Associação Brasileira de Normas Técnicas (1998), cada pacote de software deve ter uma descrição de produto. A descrição de produto é um documento que estabelece as propriedades do produto, com o propósito de orientar potenciais compradores na avaliação da adequabilidade do produto antes de comprá-lo. Caso o produto de software não disponha da descrição de produto, esta é considerada uma não-conformidade maior.

#### **3.2.1.1 REQUISITOS GERAIS SOBRE O CONTEÚDO DA DESCRIÇÃO**

Convém que a descrição do produto seja suficientemente inteligível, completa e possua boa organização e apresentação, a fim de auxiliar os compradores em potencial na avaliação da adequação do produto às necessidades, antes de comprá-lo.

Deve ser livre de inconsistências internas. Convém que cada termo tenha um único significado.

#### **3.2.1.2 IDENTIFICAÇÕES E INDICAÇÕES**

As identificações e indicações estão divididas em vários itens, são eles:

- a) identificação da descrição do produto: a descrição de produto deve possuir uma única identificação de documento. A descrição de produto pode ser chamada de formas diferentes, por exemplo, “Descrição Funcional”, “Informação de Produto” e “Folha de Produto”;
- b) identificação do produto: a descrição de produto deve identificar o produto. A identificação do produto deve ter no mínimo o nome do produto e uma versão ou data. Se existirem duas ou mais variantes mencionadas na descrição de produto, então cada variante deve ter no mínimo o nome do produto, o nome da variante e uma versão ou data;
- c) fornecedor: a descrição de produto deve conter o nome e o endereço de no mínimo um fornecedor (o nome e endereço não precisam ser impressos; o carimbo de um fornecedor/revendedor é suficiente);

- d) tarefa: a descrição de produto deve identificar as tarefas que podem ser executadas utilizando-se o produto;
- e) conformidade a documentos de requisitos: a descrição de produto pode fazer referência aos documentos de requisitos com os quais o produto está em conformidade. Neste caso as edições relevantes devem ser identificadas.
- f) requisitos de hardware e software: os requisitos para colocar o produto em uso devem ser especificados, incluindo nomes de fabricantes e identificação do tipo de todos os componentes;
- g) interface com outros produtos: se a descrição do produto faz referência a interfaces com outros produtos, as interfaces ou produtos devem ser identificados;
- h) itens a serem entregues: todo componente físico do produto fornecido deve ser identificado, em particular todos os documentos impressos e todos os meios de armazenamento de dados;
- i) instalação: deve ser declarado se a instalação do produto pode ou não ser conduzida pelo usuário;
- j) suporte: deve ser declarado se o suporte para operação do produto é oferecido ou não;
- k) manutenção: deve ser declarado se a manutenção é oferecida ou não e, se for oferecida, deve ser declarado especificamente o que é incluído.

### **3.2.1.3 DECLARAÇÕES SOBRE FUNCIONALIDADE**

A descrição de produto deve fornecer uma visão geral das funções disponíveis para o usuário do produto, os dados necessários e as facilidades oferecidas, bem como informar se o uso do produto é limitado por valores-limite específicos, e também, convém que a descrição de produto inclua informações a respeito de maneiras, se fornecidas, para evitar o acesso não autorizado a programas e dados.

### **3.2.1.4 DECLARAÇÕES SOBRE CONFIABILIDADE**

A descrição de produto deve apresentar informações sobre procedimentos para preservação de dados, ou seja, deve possuir uma declaração do tipo: “é possível fazer backup (cópia de segurança) através de funções do sistema operacional”.

Também convém que propriedades adicionais do produto sejam descritas para assegurar a capacidade funcional do mesmo, tais como: verificação se a entrada é aceitável; proteção contra conseqüências danosas decorrentes de erro de usuário; recuperação de erro; etc.

### **3.2.1.5 DECLARAÇÕES SOBRE USABILIDADE**

As declarações sobre usabilidade são:

- a) interface com o usuário: deve ser especificado o tipo de interface com o usuário. Por exemplo: se a interface com usuário é por linha de comando, por menu, por janelas, por teclas de função e função de auxílio.
- b) conhecimento requerido: deve ser descrito o conhecimento específico necessário para o seu uso. Por exemplo: conhecimento de uma determinada área técnica; conhecimento de um sistema operacional específico; conhecimento que possa ser adquirido através de treinamento especial; conhecimento de outro idioma diferente daquele que foi escrito na descrição do produto.
- c) adequação às necessidades do usuário: se o produto pode ser adaptado pelo usuário, então as ferramentas para esta adaptação e as condições para seu uso devem ser identificadas. São exemplos: mudança de parâmetros; mudança de algoritmos para computação; atribuição de teclas de função.
- d) Proteção contra infrações e direitos autorais: se a proteção técnica contra infrações a direitos autorais pode dificultar a usabilidade, então esta proteção deve ser declarada. São exemplos: proteção técnica contra cópias; datas programadas de expiração de uso; lembretes interativos para pagamento por cópia.
- e) Eficiência de uso e satisfação de usuário: a descrição do produto deve incluir dados sobre a eficiência de uso e satisfação de usuário, podendo seguir a orientação da ISO 9241-11.

### **3.2.1.6 DECLARAÇÕES SOBRE EFICIÊNCIA**

A descrição de produto pode incluir informações sobre o comportamento do produto em relação ao tempo, tais como tempo de resposta e taxas de *throughput* (processamento) para uma dada função sob condições estabelecidas.



### 3.2.1.7 DECLARAÇÕES SOBRE MANUTENIBILIDADE

A descrição de produto pode conter declarações sobre manutenibilidade, tais como: recursos necessários para o diagnóstico de deficiências ou causas de falhas.

### 3.2.1.8 DECLARAÇÕES SOBRE PORTABILIDADE

A descrição de produto pode conter declarações sobre portabilidade, ou seja, atributos de software necessários para instalar o produto nos ambientes especificados.

## 3.2.2 REQUISITOS DE QUALIDADE PARA DOCUMENTAÇÃO DE USUÁRIO

Os requisitos de qualidade para documentação do usuário formam um conjunto completo de documentos que, disponíveis em forma impressa ou não, são fornecidos para utilização de um produto de software. Tal documentação deve conter todas as informações necessárias para a instalação, para o uso e para manutenção do produto de software.

No Quadro 3-1, são definidos os requisitos de qualidade para documentação de usuário.

Excluído: Quadro 3-1

Excluído: Quadro 3-1

**Quadro 3-1 - Requisitos de qualidade para documentação de usuário**

Requisitos	Descrição
Compleitude	<p>A documentação do usuário deve conter as informações necessárias para o uso do produto. Todas as funções estabelecidas na descrição de produto e todas as funções do programa que os usuários tenham acesso devem ser completamente descritas na documentação de usuário.</p> <p>Todo valor-limite citado na descrição de produto deve ser repetido na documentação de usuário.</p> <p>Se a instalação puder ser executada pelo usuário, a documentação de usuário deverá incluir um manual de instalação contendo todas as informações necessárias. Também convém que o manual de instalação estabeleça o espaço de armazenamento mínimo para a instalação do produto.</p>

	Se algum tipo de manutenção puder ser executada pelo usuário, a documentação de usuário deverá incluir um manual de manutenção de programa contendo todas as informações necessárias para essa manutenção.
Correção	Todas as informações na documentação de usuário devem estar corretas. Além disso, convém que sua apresentação não contenha ambigüidades nem erros.
Consistência	Os documentos da documentação de usuário não devem apresentar contradições internas entre si e com a descrição de produto. Convém que cada termo possua um significado único em toda documentação.
Inteligibilidade	Convém que a documentação de usuário seja compreensível pela classe de usuários que desenvolvem atividades com o produto, utilizando termos apropriados, exibições gráficas, explicações detalhadas e citando fontes úteis de informação.
Apresentação e Organização	Convém que a documentação de usuário possua boa apresentação e organização, de tal modo que quaisquer relacionamentos seja facilmente identificados.  Convém que todo documento tenha índice analítico e remissivo.  Se um documento não estiver na forma impressa, convém que o procedimento para impressão seja identificado.

Fonte: (Sandri, 1998)

### 3.2.3 REQUISITOS DE QUALIDADE PARA PROGRAMAS E DADOS

Os requisitos de qualidade para programas e dados utilizam as mesmas definições das características da qualidade apresentadas no capítulo 3.2.1. As características de Funcionalidade, Confiabilidade e Usabilidade são destacadas e devem ser verificadas através do uso do produto. Já para as características de Eficiência, Manutenibilidade e Portabilidade, não há requisitos específicos, entretanto devem estar em conformidade com as declarações citadas na descrição de produto.

### **3.2.3.1 FUNCIONALIDADE**

Neste requisito devem ser verificados os procedimentos para instalação do produto, se há a presença de todas as funções mencionadas, se há uma execução correta das funções e se realmente há ausência de contradições entre a descrição do produto e a documentação.

- a) instalação: se a instalação puder ser realizada pelo usuário, deve ser possível instalar os programas com sucesso, seguindo as informações contidas no manual de instalação. Os requisitos de hardware e software apresentados na descrição de produto devem ser suficientes para a instalação dos programas (após a instalação deve ser possível identificar se os programas funcionam, ou usando guias de teste fornecidas ou através de auto-teste com as mensagens correspondentes).
- b) presença de funções: todas as funções mencionadas na descrição de produto e na documentação de usuário devem ser executáveis na forma nela descrita, com os correspondentes recursos, propriedades e dados, e dentro dos valores-limite fornecidos.
- c) correção: os programas e dados devem corresponder a todas as declarações contidas na descrição de produto e na documentação de usuário. As funções devem ser executadas de maneira correta para a realização de uma tarefa. Programas e dados devem estar de acordo com todos os requisitos definidos em qualquer documento de requisitos citado na descrição do produto.
- d) consistência: os programas e dados não devem conter contradições internas, contradições com a descrição do produto e com a documentação de usuário. Convém que cada termo tenha um significado único em toda a documentação. Também convém que o controle da operação do programa pelo usuário e o comportamento do programa (mensagens, formatos de tela e relatórios) sejam estruturados de maneira uniforme.

### **3.2.3.2 CONFIABILIDADE**

O sistema, compreendendo hardware e software, bem como os programas que pertencem ao produto, não deve entrar em um estado no qual o usuário não consiga controlá-lo, nem deve corromper ou perder dados, mesmo que a capacidade seja explorada até os limites especificados ou que sejam feitas tentativas para explorar a capacidade além dos limites especificados ou por uma entrada incorreta que seja feita pelo usuário ou por outros

programas listados na descrição de produto ou ainda se instruções explícitas na documentação de usuário sejam violadas.

Estão excluídas somente as possibilidades de interrupção do hardware e do sistema operacional que não podem ser controladas por nenhum programa, tal como a combinação de teclas CTRL+ALT+DEL.

Os programas devem reconhecer as violações da sintaxe estabelecida para entrada de dados. No caso de um programa reconhecer uma entrada como errônea ou indefinida ele não deve processá-la como uma entrada permitida.

### **3.2.3.3 USABILIDADE**

A comunicação entre o programa e o usuário deve ser de fácil entendimento, através das entradas de dados, mensagens e apresentação dos resultados, utilizando um vocabulário apropriado, representações gráficas e funções de auxílio, entre outras, o programa também deve proporcionar uma apresentação e organização que facilite uma visão geral das informações, além de procedimentos operacionais que o auxiliem, por exemplo, a reversão de uma função executada e o uso de recursos de hipertexto em funções de auxílio, entre outras.

Para melhor definir cada um dos requisitos da usabilidade, serão detalhados a inteligibilidade, apresentação e organização e operacionalidade.

- a) inteligibilidade: convém que as perguntas, mensagens e resultados dos programas sejam inteligíveis, por exemplo, por uma seleção adequada de termos, por representações gráficas, pelo fornecimento de informações básicas que facilitem o entendimento e pelas explicações dadas por uma função de auxílio (help).

As mensagens de erro devem fornecer informações detalhadas, explicando a sua causa ou forma de correção.

- b) apresentação e organização: cada meio de armazenamento de dado deve apresentar a identificação do produto e, se existir mais de um meio, eles devem ser distinguidos por um número ou texto.

Deve ser sempre possível ao usuário, quando estiver trabalhando com os programas, descobrir qual função está sendo executada.

Convém que os programas forneçam ao usuário informações claramente visíveis e fáceis de serem lidas. Convém que o usuário seja guiado por informações codificadas e agrupadas adequadamente. Onde for necessário, convém que os programas alertem o usuário.

Convém também que as mensagens dos programas sejam projetadas de forma que o usuário possa diferenciá-las facilmente pelo tipo, por exemplo, mensagens de confirmação, mensagens de solicitações, mensagens de advertências e mensagens de erros.

Além disso, convém que os formatos de tela de entrada, de relatórios e de outras entradas e saídas, sejam projetados para serem claros e com boa apresentação e organização. Exemplos: campos alfanuméricos sejam alinhados pela esquerda; campos numéricos sejam alinhados pela direita; limites dos campos sejam reconhecíveis; etc.

- c) operacionalidade: a execução de funções que tem conseqüências graves deve ser reversível, ou os programas devem dar uma clara advertência sobre as conseqüências e requisitar a confirmação antes da execução do comando.

#### **3.2.3.4 EFICIÊNCIA**

Não há exigência. Entretanto, o produto deve estar em conformidade com as declarações de eficiência citadas em sua descrição.

#### **3.2.3.5 MANUTENIBILIDADE**

Não há exigência. Entretanto, o produto deve estar em conformidade com as declarações de manutenibilidade citadas em sua descrição.

#### **3.2.3.6 PORTABILIDADE**

Não há exigência. Entretanto, o produto deve estar em conformidade com as declarações de portabilidade citadas em sua descrição.

### 3.2.4 INSTRUÇÕES PARA TESTE

As instruções para teste especificam como um produto deve ser testado em relação aos requisitos de qualidade. Elas incluem tanto o teste das propriedades necessárias a todos os produtos de mesmo tipo, quanto o teste das propriedades especificadas na descrição do produto.

Somente o produto, no seu ambiente de hardware e software, é testado. A avaliação ergonômica do ambiente de uso do sistema computacional não é considerada. As instruções para teste de um pacote de software são divididas em cinco fases:

- a) pré-requisitos de teste: devem estar presentes todos os componentes a serem entregues e os documentos de requisitos identificados na descrição do produto. Também, se algum tipo de treinamento for mencionado na descrição de produto, o responsável pelo teste deve ter acesso ao material e ao programa de treinamento;
- b) atividades de teste: a descrição de produto, a documentação de usuário, os programas e quaisquer dados a serem fornecidos como parte do pacote de software devem ser testados com relação à conformidade com os requisitos de qualidade já descritos neste capítulo;
- c) registros de teste: devem conter informações suficientes para permitir a repetição do teste e a identificação das pessoas envolvidas;
- d) relatório de teste: deve conter um resumo com os resultados dos testes efetuados, que deve ter a seguinte estrutura: identificação do produto; sistemas computacionais usados para o teste (hardware, software e suas configurações); documentos usados (com suas identificações); resultados dos testes de descrição de produto, documentação de usuário e programas e dados; lista das não-conformidades aos requisitos de qualidade; uma lista das não-conformidades em relação às recomendações, ou uma lista das recomendações que não foram seguidas, ou ainda uma declaração de que o produto não foi testado quanto à conformidade em relação às recomendações; e a data do encerramento do teste.
- e) teste de acompanhamento: quando um produto, que já foi testado, é testado novamente, todas as partes modificadas nos documentos, funções, dados, como também todas as partes inalteradas com possibilidade de serem influenciadas pelas partes alteradas devem ser testadas como se fossem um novo produto. Todas

as demais partes devem ser testadas, considerando-se, pelo menos, casos de testes com seleção por amostragem.

## **4 DESENVOLVIMENTO DO TRABALHO**

Neste capítulo serão descritas as atividades desempenhadas em cada fase de desenvolvimento de software adotado para construção do trabalho. Também serão relacionados e discutidos os resultados obtidos a partir do mesmo.

A seguir, será detalhada a metodologia de desenvolvimento de software adotada no presente trabalho, que possui as seguintes fases: análise dos requisitos, especificação do projeto, implementação, teste e validação.

### **4.1 ANÁLISE DOS REQUISITOS**

Nesta seção serão apresentados os principais requisitos do trabalho, através de uma contextualização do cenário que originou seu desenvolvimento, bem como as funcionalidades propostas e características gerais.

O presente trabalho caracteriza-se no desenvolvimento de uma ferramenta de software capaz de auxiliar na avaliação da qualidade de produtos de software, seja esta avaliação para fins de liberação de um produto de software no mercado ou mesmo uma comparação entre diferentes produtos de software para fins de aquisição. Identificou-se a necessidade desta ferramenta diante do aumento da competitividade entre os softwares, onde o quesito principal como forma de diferenciação em relação à concorrência é a qualidade. Portanto, desde o início do seu desenvolvimento a ferramenta objetivou tornar-se efetivamente um produto de auxílio ao processo de avaliação da qualidade de produtos de software.

A fase de análise dos requisitos foi realizada através de entrevistas informais com profissionais de algumas áreas ligadas ao assunto (consultores de software, validador de software, usuários de software) e também foram analisadas, com base nas normas ISO/IEC 9126 E NBR ISO/IEC 12119, as principais características da qualidade que um produto de software deve possuir para atender às necessidades de seus usuários. A partir destas informações, foi elaborado um questionário com uma série de questões que são a base para todo o processo de avaliação. Os requisitos de qualidade utilizados para a elaboração do questionário foram obtidos também através de trabalhos correlatos na área de qualidade de produtos de software, onde se procurou fazer uma triagem dos requisitos pesquisados nestes trabalhos, analisando cada requisito para verificar se o mesmo é pertinente à avaliação de um



produto de software de modo geral. Os trabalhos correlatos pesquisados estão demonstrados no [Quadro4-1](#).

Excluído: Quadro 4-1

Excluído: Quadro 4-1

**Quadro 4-1 – Lista de trabalhos correlatos**

Trabalhos Correlatos	Objetivo do trabalho
Protótipo de sistema para auxiliar a avaliação da qualidade de sistemas de automação comercial (Guessser, 1995).	Implementação de uma ferramenta para auxiliar a avaliação da qualidade de sistemas de automação comercial, baseado na norma ISO/IEC 9126.
Protótipo de sistema para avaliação da qualidade de softwares de folha de pagamento (Piske,1996).	Implementação de uma ferramenta para auxiliar a avaliação da qualidade de softwares de folha de pagamento, baseado na norma ISO/IEC 9126.
Protótipo para a avaliação da qualidade de softwares de contabilidade (Poffo, 1995).	Implementação de uma ferramenta para auxiliar a avaliação da qualidade de softwares de contabilidade, baseado na norma ISO/IEC 9126.
Software de apoio a avaliação da qualidade de pacotes baseado na norma ISO/IEC 12119 (Sandri, 1998).	Implementação de uma ferramenta para auxiliar a avaliação da qualidade de pacotes de software, baseado na norma NBR ISO/IEC 12119 e sua aplicação numa empresa de desenvolvimento de software para construção civil.
Proposta de avaliação da qualidade de produtos de software utilizando a norma ISO/IEC 9126 (Storch, 2000).	Propor um roteiro de avaliação da qualidade de produtos de software baseado na norma ISO/IEC 9126.

Através dos requisitos mencionados e do detalhamento das normas ISO/IEC 9126 e NBR ISO/IEC 12119, tornou-se possível o desenvolvimento da ferramenta para avaliação da qualidade de produtos de software, conforme definido no objetivo do trabalho.

## 4.2 ESPECIFICAÇÃO

Esta seção descreve as atividades desempenhadas na fase de especificação do projeto, apresentando os modelos, diagramas e estruturas de dados que constituem na representação lógica do trabalho.

Utilizou-se para a especificação do protótipo a ferramenta CASE Power Designer 6.1, que 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 e documentação de dados existentes.

O Power Designer 6.1 trabalha em várias plataformas de comunicação a qual o sistema operacional 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.

Em relação à portabilidade, o Power Designer 6.1, possibilita a comunicação com alguns softwares conhecidos como Delphi e Visual Basic. Através da facilidade de customização, a ferramenta atende a várias fases do ciclo de desenvolvimento. A sua flexibilidade de configuração o torna abrangente, modelando informações às necessidades da metodologia utilizada.

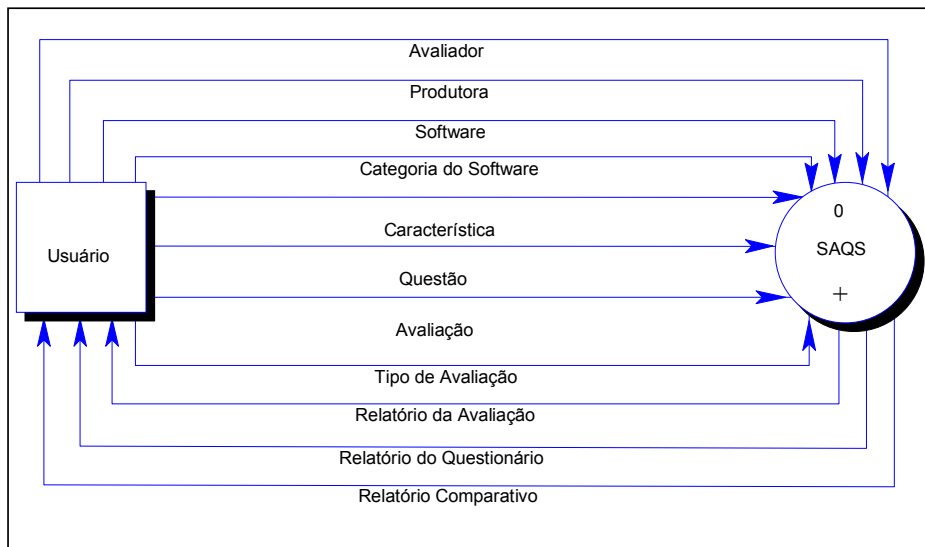
### 4.2.1 DIAGRAMA DE CONTEXTO

O diagrama de contexto identifica uma visão macro do sistema como um todo, declarando o objetivo principal do sistema num só processo. Na [Figura 4-1](#), é apresentado o diagrama de contexto, utilizando aqui a notação de Yourdon, disponível na própria ferramenta CASE.

Excluído: Figura 4-1

Excluído: Figura 1-1

**Figura 4-1 - Diagrama de contexto**



Excluído: 4

Excluído: 1

O sistema de avaliação da qualidade de software (SAQS) evidenciado na [Figura 4-1](#), é composto de uma entidade externa chamada usuário. O usuário é a pessoa que entrará com as informações necessárias para que a avaliação do produto de software seja processada. Ele terá como informações de retorno, relatórios das avaliações já efetuadas, relatório dos questionários e também um relatório comparativo de avaliações.

Excluído: Figura 4-1

Excluído: Figura 1-1

## 4.2.2 DIAGRAMA DE FLUXO DE DADOS (DFD)

O diagrama de fluxo de dados descreve graficamente o sentido do fluxo das informações no sistema, identificando as entidades externas com as quais ele se comunica. As entidades em conjunto com o sentido do fluxo das informações apresentam uma visão geral do comportamento do sistema na perspectiva do usuário, e também, na perspectiva do desenvolvedor, apresenta uma forma lógica e com um nível de detalhamento suficiente para o desenvolvimento do sistema.

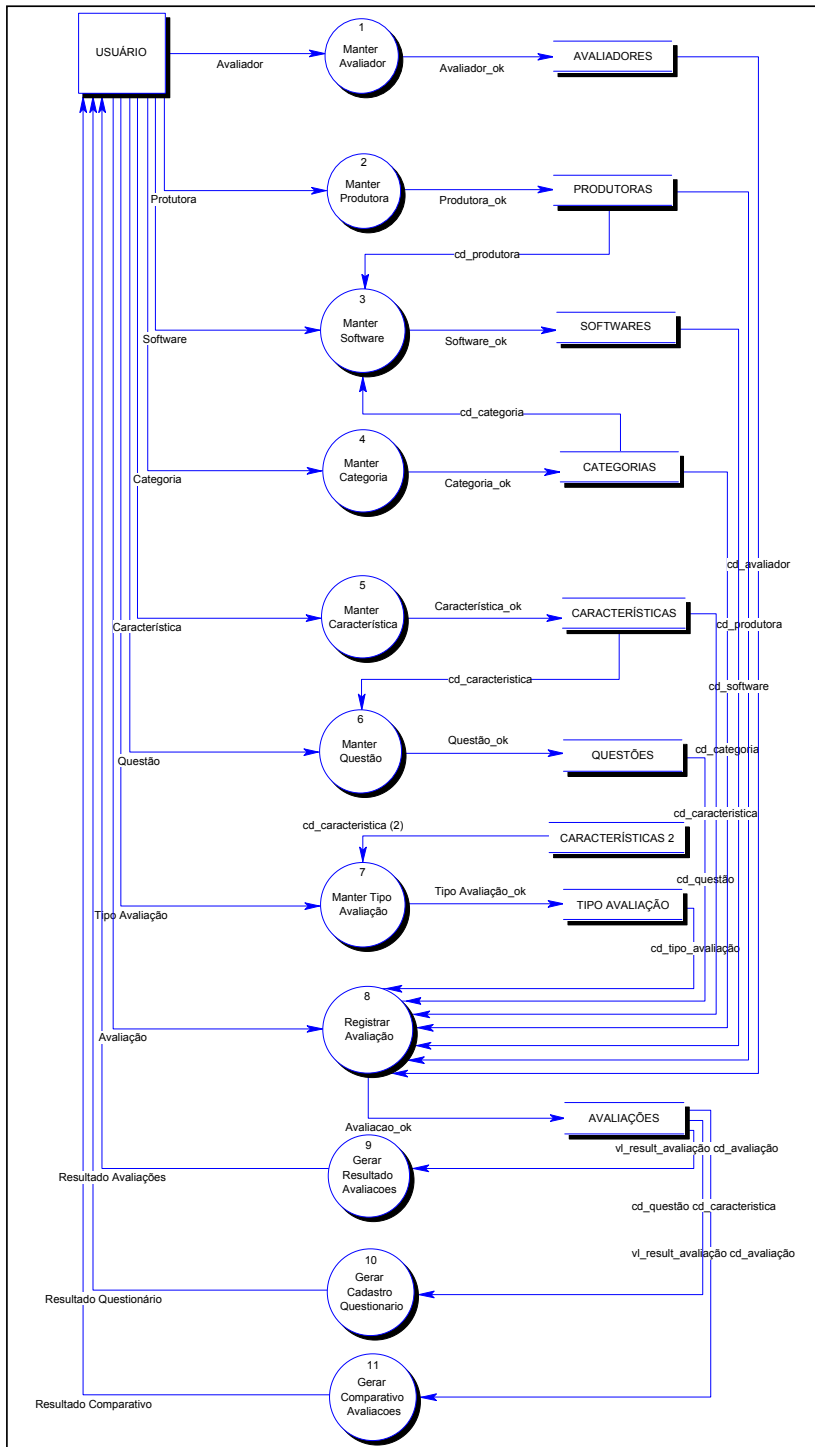
Na [Figura 4-2](#), está representado o diagrama de fluxo de dados.

Excluído: Figura 4-3

Excluído: Figura 1-3

Figura 4-2 - Diagrama de fluxo de dados

- Excluído: 4
- Excluído: 1
- Excluído: 3



No diagrama de fluxo de dados referenciado na [Figura 4-2](#), o usuário é quem fará entradas e receberá os resultados em forma de relatório. Para que seja possível atender todas as necessidades do usuário, foram criados os seguintes processos:

Excluído: Figura 4-3

Excluído: Figura 1-3

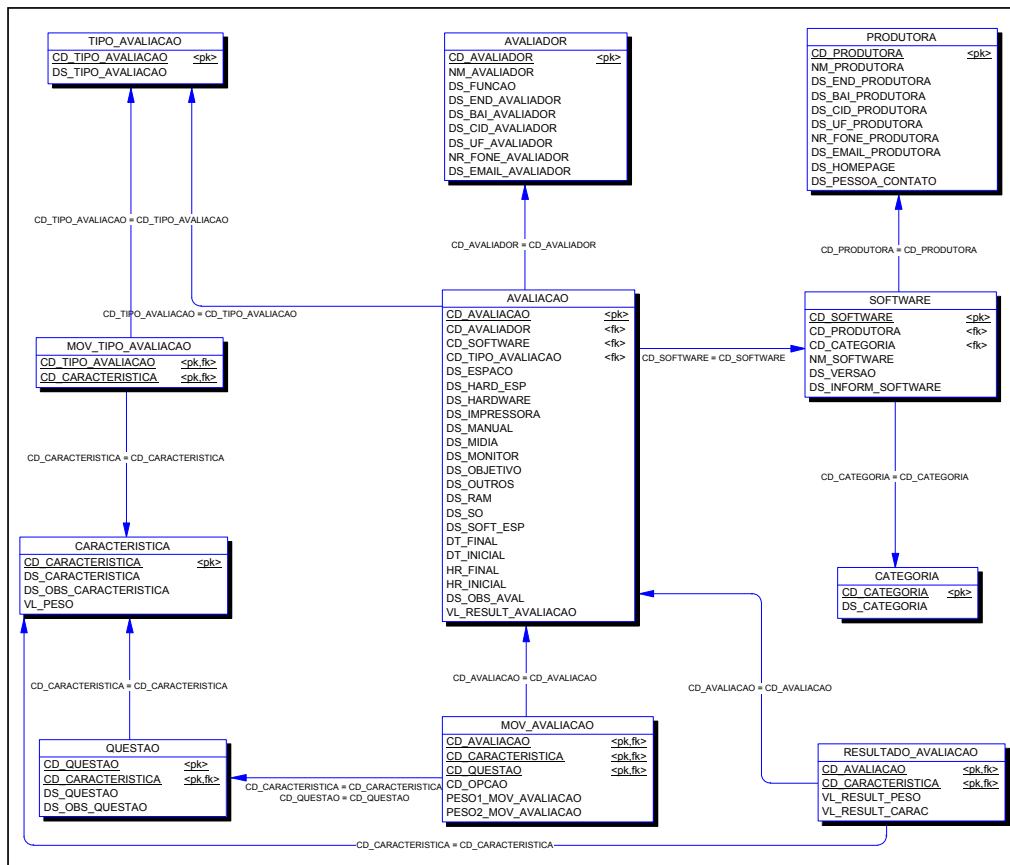
- a) processos de 1.1 a 1.6: representam as informações cadastrais da ferramenta. Através destes cadastros será possível avançar para as próximas etapas da ferramenta. As entidades utilizadas para estes processos são: Avaliadores, Produtoras, Softwares, Categorias (software), Características e Questões.
- b) processo 1.7: trata da configuração relativa às avaliações. Neste ponto são definidos os tipos de avaliações que poderão ser efetuados pelo usuário onde o mesmo poderá escolher quais características da qualidade deseja avaliar. Por este motivo, a entidade Tipo Avaliação faz referência à entidade Características, além da criação de uma entidade dependente, utilizada para armazenagem de dados de movimentação do tipo de avaliação.
- c) processo 1.8: trata da entrada inicial para o processo de avaliação de produtos de software onde são utilizadas todas as entidades de cunho cadastral citadas nos itens anteriores. Portanto este processo receberá diversas informações não só cadastrais, mas também informações vindas diretamente da entidade externa chamada Usuário. Exemplos de informações recebidas por este processo: código da avaliação, objetivo da avaliação, data inicial e final da avaliação, código do avaliador, código do software, entre outros. Esta entidade, Avaliações, também armazena informação relativa à opção feita pelo usuário no momento de avaliar um determinado requisito da qualidade. Além disso, ainda há a criação de uma entidade dependente, utilizada para armazenagem de dados de movimentação da avaliação.
- d) processo 1.9: gera relatório da avaliação. Para este processo, serão utilizadas todas as entidades de movimentação em conjunto com as entidades de cadastro, montando desta forma um espelho da avaliação de forma analítica, informando o resultado da avaliação no final do relatório.
- e) processo 1.10: este processo gera relatório cadastral das questões. Utilizará as entidades de cadastro Características e Questões, mas dependerá da entidade Tipo Avaliação, pois é a partir desta entidade é que serão escolhidas quais as características e questões a serem impressas no relatório.

- f) processo 1.11: gera relatório comparativo entre duas avaliações, utilizando-se das mesmas entidades do processo 1.9.

### 4.2.3 DIAGRAMA ENTIDADE RELACIONAMENTO (DER)

O diagrama ou modelo entidade relacionamento (DER), enfatiza os principais objetivos ou entidades do sistema. Detalha as associações existentes entre as entidades de dados utilizando componentes semânticos próprios. O Modelo entidade Relacionamento é apresentado na [Figura 4-3](#).

**Figura 4-3 - Modelo entidade relacionamento**



Excluído: Figura 4-5  
 Excluído: Figura 1-5  
 Excluído: 4  
 Excluído: 1  
 Excluído: 5

No modelo entidade relacionamento representado na [Figura 4-3](#), todas as entidades possuem obrigatoriedade com relação à entidade principal, permitindo um relacionamento de

Excluído: Figura 4-5  
 Excluído: Figura 1-5

1 para N ou 1 para 1, ou seja, torna-se obrigatório existir um registro pai para que se cadastre um ou mais registros na entidade filho.

A avaliação é realizada sobre produtos de software ligados a uma produtora de software e a uma categoria de software, desta forma, é necessário que haja um produto de software cadastrado para que se informe qual produto de software será avaliado. Ainda, para ser realizada a avaliação, torna-se também obrigatório o cadastramento de pelo menos uma característica da qualidade e conseqüentemente pelo menos uma questão para esta característica, para que a mesma possa ser avaliada. Outro dado necessário para a avaliação é o avaliador, que entrará com as respostas à avaliação e também um tipo de avaliação que na verdade é a configuração de como a avaliação será conduzida.

A configuração da avaliação que se dá através do tipo de avaliação, constitui na montagem ou seleção das características da qualidade que se deseja aplicar para uma determinada avaliação, desta forma torna-se obrigatório cadastrar ao menos uma questão para a característica a ser avaliada. Para estas movimentações há tabelas auxiliares de movimentação, onde ficam armazenados os dados referentes a cada movimento, conforme a estrutura da avaliação.

Por fim, é montado o resultado de cada característica da qualidade na tabela Resultado\_Avaliação e com estes dados calculado o resultado geral da avaliação, gravando-o na tabela Avaliação.

#### 4.2.4 DICIONÁRIO DE DADOS

**O dicionário de dados contém a definição de todos os dados mencionados no DER, 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 Power**

**Designer, utilizada para o desenvolvimento da modelagem. A seguir, do Quadro 4-2, ao**

**Quadro 4-12, estão descritas as entidades com seus respectivos atributos (nome, código, tipo, se o atributo é chave primária e se o atributo é obrigatório):**

Excluído: Quadro 4-3

Excluído: Quadro 1-3

Excluído: ¶  
Quadro 4-14

Excluído: ¶  
Quadro 1-14

Quadro 4-2 - Entidade de avaliações (dicionário de dados)

AVALIACAO				
Name	Code	Type	P	M
CODIGO AVALIACAO	CD_AVALIACAO	Integer	Yes	Yes
CODIGO AVALIADOR	CD_AVALIADOR	Integer	No	Yes
CODIGO SOFTWARE	CD_SOFTWARE	Integer	No	Yes
CODIGO TIPO AVALIACAO	CD_TIPO_AVALIACAO	Integer	No	Yes
ESPACO HD	DS_ESPACO	char(10)	No	No
HARDWARE ESPECIAL	DS_HARD_ESP	char(60)	No	No
HARDWARE	DS_HARDWARE	char(60)	No	No
IMPRESSORA	DS_IMPRESSORA	char(20)	No	No
MANUAIS	DS_MANUAL	char(60)	No	No
MIDIA	DS_MIDIA	char(60)	No	No
MONITOR	DS_MONITOR	char(20)	No	No
OBJETIVO	DS_OBJETIVO	char(60)	No	No
OUTROS	DS_OUTROS	char(60)	No	No
MEMORIA RAM	DS_RAM	char(10)	No	No
SISTEMA OPERACIONAL	DS_SO	char(20)	No	No
SOFTWARE ESPECIAL	DS_SOFT_ESP	char(20)	No	No
DATA FINAL	DT_FINAL	date	No	No
DATA INICIAL	DT_INICIAL	date	No	No
HORA FINAL	HR_FINAL	time	No	No
HORA INICIAL	HR_INICIAL	time	No	No
VALOR RESULTADO AVALIACAO	VL_RESULT_AVALIACAO	decimal(3,2)	No	No
OBSERVACAO AVALIACAO	DS_OBS_AVAL	char(60)	No	No

Excluído: 4

Excluído: 1

Excluído: 3



**Quadro 4-3 - Entidade de avaliadores (dicionário de dados)**

Excluído: 4

Excluído: 1

AVALIADOR				
Name	Code	Type	P	M
CODIGO AVALIADOR	CD_AVALIADOR	integer	Yes	Yes
NOME AVALIADOR	NM_AVALIADOR	char(30)	No	No
FUNCAO	DS_FUNCAO	char(30)	No	No
ENDERECO AVALIADOR	DS_END_AVALIADOR	char(30)	No	No
BAIRRO AVALIADOR	DS_BAI_AVALIADOR	char(15)	No	No
CIDADE AVALIADOR	DS_CID_AVALIADOR	char(20)	No	No
UF AVALIADOR	DS_UF_AVALIADOR	char(2)	No	No
FONE AVALIADOR	NR_FONE_AVALIADOR	char(15)	No	No
EMAIL AVALIADOR	DS_EMAIL_AVALIADOR	char(30)	No	No

**Quadro 4-4 - Entidade de características (dicionário de dados)**

Excluído: 4

Excluído: 1

CARACTERISTICA				
Name	Code	Type	P	M
CODIGO CARACTERISTICA	CD_CARACTERISTICA	char(5)	Yes	Yes
DESCRICAO CARACTERISTICA	DS_CARACTERISTICA	char(50)	No	No
OBSERVACAO CARACTERISTICA	DS_OBS_CARACTERISTICA	char(200)	No	No
PESO	VL_PESO	decimal(3,2)	No	No

**Quadro 4-5 - Entidade de categorias (dicionário de dados)**

Excluído: 4

Excluído: 1

CATEGORIA				
Name	Code	Type	P	M
CODIGO CATEGORIA	CD_CATEGORIA	integer	Yes	Yes
DESCRICAO CATEGORIA	DS_CATEGORIA	char(30)	No	No

**Quadro 4-6 - Entidade de movimentos de avaliação (dicionário de dados)**

Excluído: 4

Excluído: 1

MOV_AVALIAÇÃO				
Name	Code	Type	P	M
CODIGO AVALIACAO	CD_AVALIACAO	integer	Yes	Yes
CODIGO CARACTERISTICA	CD_CARACTERISTICA	char(5)	Yes	Yes
CODIGO QUESTAO	CD_QUESTAO	integer	Yes	Yes
OPCAO	CD_OPCAO	integer	No	No
PESO1 MOV AVALIACAO	PESO1_MOV_AVALIACAO	decimal(3,2)	No	No
PESO2 MOV AVALIACAO	PESO2_MOV_AVALIACAO	decimal(3,2)	No	No

**Quadro 4-7 - Entidade de movimento dos tipos de avaliação (dicionário de dados)**

Excluído: 4

Excluído: 1

MOV_TIPO_AVALIACAO				
Name	Code	Type	P	M
CODIGO TIPO AVALIACAO	CD_TIPO_AVALIACAO	integer	Yes	Yes
CODIGO CARACTERISTICA	CD_CARACTERISTICA	char(5)	Yes	Yes

**Quadro 4-8 - Entidade de questões (dicionário de dados)**

Excluído: 4

Excluído: 1

QUESTAO				
Name	Code	Type	P	M
CODIGO QUESTAO	CD_QUESTAO	integer	Yes	Yes
CODIGO CARACTERISTICA	CD_CARACTERISTICA	char(5)	Yes	Yes
DESCRICAO QUESTAO	DS_QUESTAO	char(200)	No	No
OBSERVACAO QUESTAO	DS_OBS_QUESTAO	char(200)	No	No

**Quadro 4-9 - Entidade de produtoras (dicionário de dados)**

Excluído: 4

Excluído: 1

PRODUTORA				
Name	Code	Type	P	M
CODIGO PRODUTORA	CD_PRODUTORA	integer	Yes	Yes

PRODUTORA				
Name	Code	Type	P	M
NOME PRODUTORA	NM_PRODUTORA	char(50)	No	No
ENDERECO PRODUTORA	DS_END_PRODUTORA	char(50)	No	No
BAIRRO PRODUTORA	DS_BAI_PRODUTORA	char(30)	No	No
CIDADE PRODUTORA	DS_CID_PRODUTORA	char(20)	No	No
UF PRODUTORA	DS_UF_PRODUTORA	char(2)	No	No
FONE PRODUTORA	NR_FONE_PRODUTORA	char(15)	No	No
EMAIL PRODUTORA	DS_EMAIL_PRODUTORA	char(30)	No	No
HOME PAGE	DS_HOMEPAGE	char(30)	No	No
PESSOA CONTATO	DS_PESSOA_CONTATO	char(20)	No	No

Quadro 4-10 - Entidade de resultado da avaliação (dicionário de dados)

Excluído: 4

Excluído: 1

RESULTADO AVALIACAO				
Name	Code	Type	P	M
CODIGO AVALIACAO	CD_AVALIACAO	integer	Yes	Yes
CODIGO CARACTERISTICA	CD_CARACTERISTICA	char(5)	Yes	Yes
RESULTADO PESO CARAC	VL_RESULT_PESO	decimal(3,2)	No	No
RESULTADO CARACTERISTICA	VL_RESULT_CARAC	decimal(3,2)	No	No

Quadro 4-11 - Entidade de tipos de avaliações (dicionário de dados)

Excluído: 4

Excluído: 1

TIPO_AVALIACAO				
Name	Code	Type	P	M
CODIGO TIPO AVALIACAO	CD_TIPO_AVALIACAO	integer	Yes	Yes
DESCRICAO TIPO AVALIACAO	DS_TIPO_AVALIACAO	char(30)	No	No

**Quadro 4-12 - Entidade de softwares (dicionário de dados)**

SOFTWARE				
Name	Code	Type	P	M
CODIGO SOFTWARE	CD_SOFTWARE	integer	Yes	Yes
CODIGO PRODUTORA	CD_PRODUTORA	integer	No	Yes
CODIGO CATEGORIA	CD_CATEGORIA	integer	No	Yes
NOME SOFTWARE	NM_SOFTWARE	char(60)	No	No
VERSAO	DS_VERSAO	char(10)	No	No
INFORMACOES SOFTWARE	DS_INFORM_SOFTWARE	char(200)	No	No

Excluído: 4

Excluído: 1

Excluído: 14

### 4.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as principais telas disponíveis para utilização passo a passo da ferramenta. Mas antes de apresentá-las, se faz necessário o entendimento do funcionamento das métricas utilizadas para se chegar ao resultado final que é o nível de qualidade do produto de software avaliado.

A ferramenta tem por objetivo avaliar a qualidade de produtos de software de qualquer área, dando suporte às empresas na melhoria da qualidade de seus produtos, visando a satisfação do cliente e o melhoramento contínuo. A ferramenta se propõe também a ajudar o usuário a fazer a escolha do software que mais lhe convém diante de diversos produtos de software de mesma categoria. Segundo a norma ISO/IEC 9126 (ABNT, 1996), 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. Baseado nesta afirmação, para a ferramenta desenvolvida foi estabelecido um modelo próprio para o processo de avaliação que será explicado logo adiante.

A ferramenta contém o cadastro da estrutura de características da qualidade conforme a montagem do modelo entidade relacionamento. Cada característica será avaliada de acordo com seu peso que está cadastrado, podendo também ser alterado pelo avaliador no momento

da avaliação. As características também podem ser alteradas, ou mesmo, inseridas novas características, bem como as perguntas do questionário de avaliação, tornando a avaliação ainda mais criteriosa (as questões que foram implementadas como exemplo na ferramenta encontram-se no Apêndice A – Listagem do *checklist* desenvolvido a partir das normas ISO/IEC 9126 E NBR ISO/IEC 12119).

Através das respostas às questões referentes a cada característica é calculado o nível de qualidade de cada característica, se a resposta dada for “Não Atende”, significa que não há evidência da existência do atributo definido no software, portanto, esta questão terá nota 1. Se a resposta dada for “Atende Parcialmente”, significa que há alguma evidência da existência do atributo definido, mas não atende satisfatoriamente aos requisitos, portanto, terá nota 2. Se a resposta dada for “Atende Largamente”, significa que há significativa evidência da existência do atributo definido, e alguma irregularidade existente não compromete o produto, portanto, terá nota 3. E por fim, se a resposta dada for “Atende Completamente”, significa que o atributo definido atende inteiramente aos requisitos e não apresenta nenhuma irregularidade, portanto, terá nota 4.

A fórmula para o cálculo do nível de pontuação ou a qualidade da característica, se faz sempre da característica mais interna (subcaracterística) para a característica mais externa (característica), ou seja, o cálculo começa pela característica que possui no segundo nível de seu código, o código diferente de “00”, exemplo 01.01 – Adequação. Após ter calculado o nível de todas as subcaracterísticas, por assim dizer, que fazem parte do nível mais externo do código de uma característica, serão obtidos os resultados destas subcaracterísticas para calcular o nível de pontuação da característica mais externa, ou seja, que possui “00” no segundo nível do código da característica, exemplo, 01.00 – Funcionalidade. Por este motivo, foi estabelecido que as questões estarão diretamente ligadas a uma subcaracterística, ou seja, à uma característica em que o segundo nível de seu código seja diferente de “00”.

A fórmula para o cálculo do nível de qualidade da subcaracterística é a seguinte:

$$NQSC = \frac{\sum N * P}{\sum N M}$$

ou seja,

calcula-se a média ponderada de cada subcaracterística: somam-se os valores das notas dadas a cada questão referente a uma subcaracterística, multiplica-se pelo peso desta subcaracterística, dividindo tudo pela soma das notas máximas atribuídas para cada questão. Esta nota máxima atribuída para cada questão, significa dizer que todas as questões poderão ter como resposta, por exemplo, “Atende Completamente”, recebendo a nota 4. Simplificando, conta-se o número de questões e multiplica-se pela nota máxima (4), para obter o somatório das notas máximas.

### 4.3.1 ESTUDO DE CASO

Para o melhor entendimento do funcionamento da fórmula, será apresentado um exemplo prático do cálculo do nível de qualidade da característica através de um estudo de caso.

Estudo de caso: avaliar um dado produto de software do ramo financeiro no intuito de se saber o nível de qualidade deste produto, informando ao final da avaliação se o mesmo pode ser considerado um produto de software de boa qualidade ou não. A avaliação do software financeiro deve ser feita de acordo com as características e subcaracterísticas da qualidade e respectivos pesos abaixo citados e para cada subcaracterística deve ser efetuada ao menos uma questão que evidencie o requisito de qualidade solicitado:

01.00 – Funcionalidade, que possui um peso de valor 60

01.01 – Adequação, que possui um peso de valor 25

01.02 – Acurácia, que possui um peso de valor 30

02.00 – Confiabilidade, que possui um peso de 40

02.01 – Maturidade, que possui um peso de 15

Com base nas informações acima, a avaliação se dá da seguinte maneira:

**Característica** : 01.00 – Funcionalidade **Peso** : 60

**Subcaracterística** : 01.01 – Adequação **Peso** : 25

**Questões:**

001 - O software possui todas as funções no produto?

( ) Não Atende      ( ) Parcialmente      ( ) Largamente      (x) Completamente

002 - O software dispõe de todas as funções necessárias para a sua execução?

Não Atende       Parcialmente       Largamente       Completamente

003 - O software dispõe de funções para processamento de rotinas específicas?

Não Atende       Parcialmente       Largamente       Completamente

**Subcaracterística** : 01.02 – Acurácia      **Peso** : 30

**Questões:**

001 - O software é preciso na execução das funções?

Não Atende       Parcialmente       Largamente       Completamente

002 - O software é preciso nos resultados?

Não Atende       Parcialmente       Largamente       Completamente

**Característica** : 02.00 – Confiabilidade      **Peso** : 40

**Subcaracterística** : 02.01 – Maturidade      **Peso** : 15

**Questões:**

001 - O software tem capacidade de continuar executando na ocorrência de erros de execução?

Não Atende       Parcialmente       Largamente       Completamente

002 - O software tem capacidade de continuar executando na ocorrência de erros do usuário?

Não Atende       Parcialmente       Largamente       Completamente

Resultado da avaliação de cada subcaracterística:

**Característica** : 01.00 – Funcionalidade

**Subcaracterística** : 01.01 – Adequação

**Soma das notas** : 12

**Peso** : 25

**Soma máxima das notas** : 12

**NQSC da Adequação** :  $(12 * 25) / 12 = 25,00$

**Subcaracterística** : 01.02 – Acurácia

**Soma das notas** : 5

**Peso** : 30

<b>Soma máxima das notas</b>	: 8
<b>NQSC da Acurácia</b>	: $(5 * 30) / 8 = 18,75$
<b>Característica</b>	: 02.00 – Confiabilidade
<b>Subcaracterística</b>	: 02.01 – Maturidade
<b>Soma das notas</b>	: 6
<b>Peso</b>	: 15
<b>Soma máxima das notas</b>	: 8
<b>NQSC da Maturidade</b>	: $(6 * 15) / 8 = 11,25$

Após obter o resultado de cada subcaracterística, parte-se para o cálculo do nível de qualidade de cada característica com a seguinte fórmula:

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

ou seja,

**NQC Funcionalidade** =  $(25,00 + 18,75) / (25 + 30) = 0,7954$  ou 79,54% que, aplicando a regra de três, corresponde ao valor de 47,73 em relação ao peso da característica Funcionalidade que é 60, portanto,

$$\text{NQC Funcionalidade} = 47,73$$

**NQC Confiabilidade** =  $11,25 / 15 = 0,75$  ou 75% que, aplicando a regra de três, corresponde ao valor de 30 em relação ao peso da característica Confiabilidade que é 40, portanto,

$$\text{NQC Confiabilidade} = 30$$

Chegando-se ao resultado de cada característica, se faz necessário calcular a qualidade final do produto, sendo necessário a seguinte fórmula:

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



onde,

P é o peso das características, ou seja, descrevendo a fórmula:

$$\text{QFP} = \frac{\text{Nível de Qualidade Funcionalidade} + \text{Nível Qualidade Confiabilidade} + \dots}{\text{Peso Funcionalidade} + \text{Peso Confiabilidade} + \dots}$$

aplicando a fórmula ao exemplo temos:

$$\text{QFP} = (47,73 + 30) / (60 + 40) = 0,7773 \text{ ou } 77,73\%$$

Chegando-se ao final da avaliação, são demonstrados os valores obtidos para cada característica e subcaracterística da qualidade, conforme o [Quadro4- 13](#).

Excluído: Quadro 4-13

Excluído: Quadro 1-13

Excluído: 4

Excluído: 1

**Quadro 4-13 - Resultado da Avaliação**

Característica	Peso	Resultado (Valor)	Resultado (%)
01.00 – Funcionalidade	60	47,73	79,55
01.01 – Adequação	25	25,00	100,00
01.02 – Acurácia	30	18,75	62,50
02.00 – Confiabilidade	40	30,00	75,00
02.01 – Maturidade	15	11,25	75,00
<b>RESULTADO FINAL</b>			<b>77,73</b>

Para o julgamento do resultado da avaliação, compara-se o resultado obtido na QFP com as faixas definidas, conforme ISO/IEC 9126 (NBR, 1996), podendo assumir os valores:

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

Portanto, de acordo com o exemplo dado, o resultado ficou na faixa de 75,6% a 90,5%, o que quer dizer que o produto de software avaliado é **Bom**, ou seja, é aceito com um produto de qualidade. Neste caso, seria interessante verificar quais são as características da qualidade

mais relevantes para esta categoria de produto de software, para que, analisando cada resposta dada às questões destas características, se possa observar em que situações o software não se comporta de forma adequada, para que seja melhorado justamente nestes pontos.

### 4.3.2 DESCRIÇÃO DAS TELAS DA FERRAMENTA

Ao executar o aplicativo, será apresentada a tela principal da ferramenta, disponibilizando acesso aos recursos da mesma, conforme a [Figura 4-4](#).

**Figura 4-4 - Tela principal do sistema**



Excluído: Figura 4-7

Excluído: Figura 1-7

Excluído: 4

Excluído: 1

Excluído: 7

O primeiro passo para o usuário utilizar a ferramenta, é efetuar o cadastramento das tabelas, o mesmo deverá acessar o item do menu chamado “Cadastros”, nele estão as telas de manutenção dos cadastros da ferramenta: avaliadores, produtoras, softwares, categorias, características e questões. São nessas opções que o usuário irá armazenar os dados relevantes para a utilização da avaliação de produtos de software.

Após o cadastramento das tabelas, para que o usuário possa trabalhar com a movimentação de avaliações, deverá ser acessado o item do menu chamado “Movimentos”. Nele estão as telas de tipos de avaliação e avaliação. No submenu “Tipos de Avaliação”, o usuário deve configurar a avaliação, informando quais características da qualidade deseja

avaliar para posteriormente acessar o submenu “Avaliação” onde o usuário entrará com as respostas solicitadas às questões de cada característica escolhida no tipo de avaliação.

Para a emissão de relatórios, o usuário encontra no terceiro item de menu, a opção “Relatórios”, com as principais saídas da ferramenta, que são: relatório da avaliação, relatório comparativo e avaliação em branco.

Quando se deseja fazer avaliação de um determinado produto, não é necessário que o avaliador ou a pessoa interessada em fazer a avaliação saiba os pesos que devem possuir cada característica da qualidade, isso porque a ferramenta já traz no cadastro das características um peso padrão para cada uma delas. A ferramenta tem como padrão de peso da característica a faixa aplicada na tabela de avaliação de produtos de software – parte 4: via do comprador (ISO, 1995), que adota o seguinte padrão:

Funcionalidade	=	Alto
Confiabilidade	=	Baixo
Usabilidade	=	Alto
Eficiência	=	Médio
Manutenibilidade	=	Baixo

Quanto ao peso das subcaracterísticas, os mesmos foram definidos por convenção do formando.

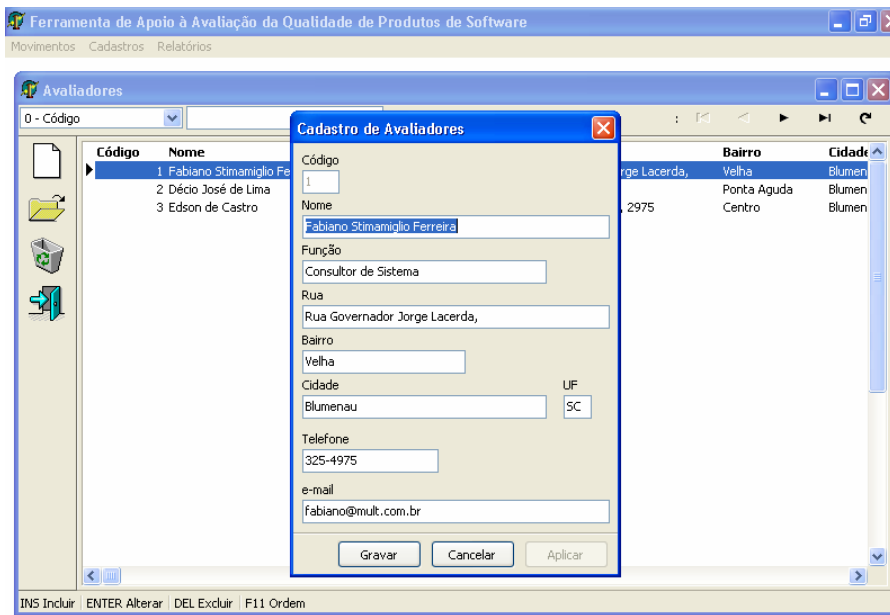
Os passos e as atividades que o usuário/avaliador executa na avaliação do software são descritos a seguir:

- a) cadastro de avaliadores: na [Figura 4-5](#), está representado o cadastro de avaliadores em que o usuário/avaliador deverá entrar com os dados do avaliador, que posteriormente será necessário informar no momento da avaliação. Nesta tela também são executadas as manutenções do cadastro de avaliadores.

**Excluído:** Figura 4-9

**Excluído:** Figura 1-9

**Figura 4-5 – Tela de cadastro de avaliadores**



Excluído: 4

Excluído: 1

Excluído: 9

- b) cadastro de produtoras: na [Figura 4-6](#), está representado o cadastro de produtoras em que o usuário/avaliador deverá entrar com os dados da produtora, que posteriormente será necessário informar no momento de cadastrar o produto de software, por este motivo, a produtora deve ser cadastrada anteriormente ao cadastro do produto de software. Nesta tela também são executadas as manutenções do cadastro de produtoras;

Excluído: Figura 4-11

Excluído: Figura 1-11

- c) cadastro de software: na [Figura 4-7](#), está representado o cadastro de softwares em que o usuário/avaliador deverá entrar com os dados do software, que posteriormente será necessário informar no momento de executar a avaliação. Ainda no cadastro do software são associadas a produtora do software e a categoria. Nesta tela também são executadas as manutenções do cadastro do software;

Excluído: Figura 4-13

Excluído: Figura 1-13

**Figura 4-6 – Tela de cadastro de produtoras**

Ferramenta de Apoio à Avaliação da Qualidade de Produtos de Software  
Movimentos Cadastros Relatórios

Produtoras

0 - Código

Código	Nome
1	Mult Sistemas Ltda
2	Benner Sistemas
3	Senior Sistemas

Produtora

Código: 1

Nome: Mult Sistemas Ltda

Endereço: Rua dos Incas, 59

Bairro: Ponta Aguda

Cidade: Blumenau

UF: SC

Fone: 326-9400

e-mail: suporte@mult.com.br

Home Page: www.mult.com.br

Contato: Fabiano

Gravar Cancelar Aplicar

INS Incluir ENTER Alterar DEL Excluir F11

Excluído: 4

Excluído: 1

Excluído: 11

**Figura 4-7 – Tela de cadastro de softwares**

Ferramenta de Apoio à Avaliação da Qualidade de Produtos de Software  
Movimentos Cadastros Relatórios

Softwares

0 - Código

Código	Nome	Versão	Categoria	Produtora
1	Máximo Finanças	6.0c		
2	Max Emp			
3	Sapiens			
4	Benner T			
5	Máximo			

Software

Código: 1

Nome: Máximo Finanças

Versão: 6.0c

Informações: Software financeiro com diversos recursos, tais como: Controle de Contas a Receber, Contas a Pagar, Tesouraria, Fluxo de Caixa, etc...

Categoria: 4 Financeiro

Produtora: 1 Mult Sistemas Ltda

Gravar Cancelar Aplicar

Informações: Software financeiro com diversos recursos, tais como: Controle de Contas a Receber, Contas a Pagar, Tesouraria, Fluxo de Caixa, etc...

Categoria: Financeiro

Produtora: Mult Sistemas Ltda

INS Incluir ENTER Alterar DEL Excluir F6 Detalhes F11 Ordem

Excluído: 4

Excluído: 1

Excluído: 13

- d) cadastro de categorias: na [Figura 4-8](#) está representado o cadastro de categorias em que o usuário/avaliador deverá entrar com os dados da categoria, que posteriormente será necessário informar no momento de cadastrar o produto de software, por este motivo, a categoria deve ser cadastrada anteriormente ao cadastro do produto de software. Nesta tela também são executadas as manutenções do cadastro de categorias;
- e) cadastro de características: na [Figura 4-9](#), está representado o cadastro de características em que o usuário/avaliador deverá entrar com os dados das características e subcaracterísticas da qualidade, que posteriormente serão necessárias informar no momento de configurar a avaliação, nos tipos de avaliação. No cadastro da característica/subcaracterística é permitido fazer observações para o melhor entendimento da mesma. Nesta tela também são executadas as manutenções do cadastro da característica/subcaracterística;

Excluído: Figura 4-15

Excluído: Figura 1-15

Excluído: Figura 4-17

Excluído: Figura 1-17

**Figura 4-8** – Tela de cadastro de categorias

Excluído: 4

Excluído: 1

Excluído: 15

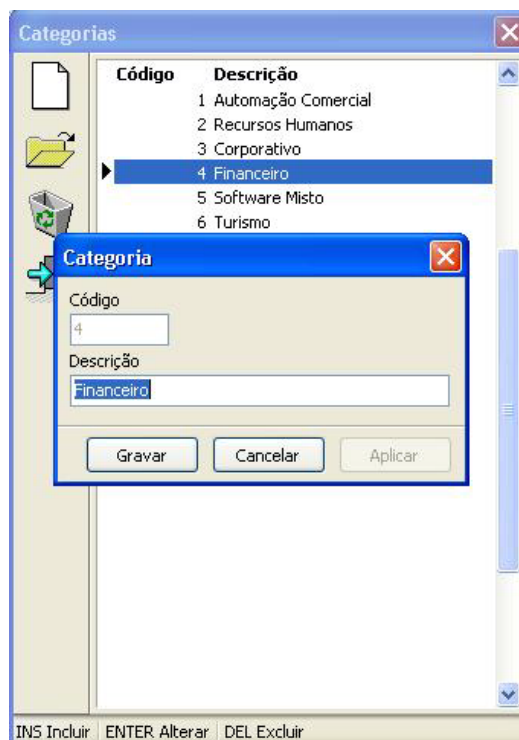
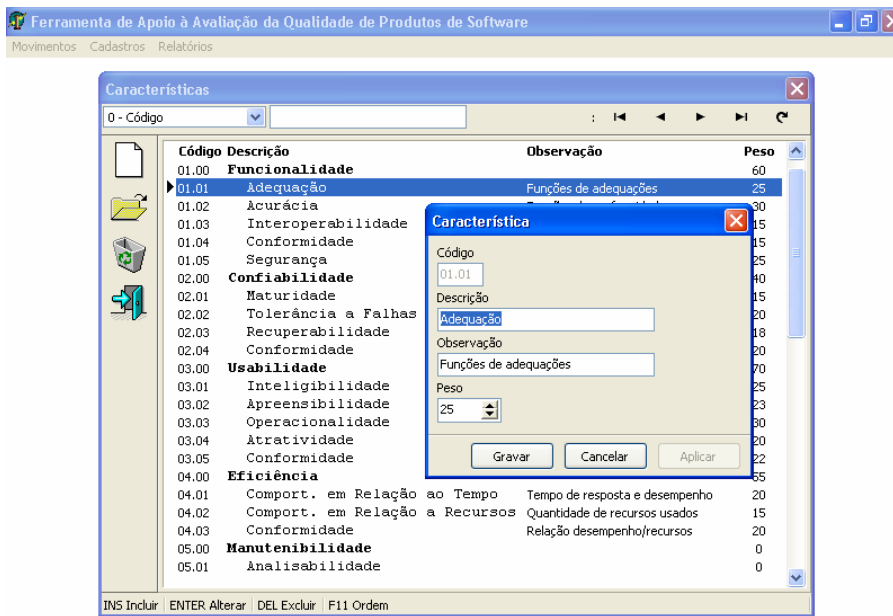


Figura 4-9 – Tela de cadastro de características

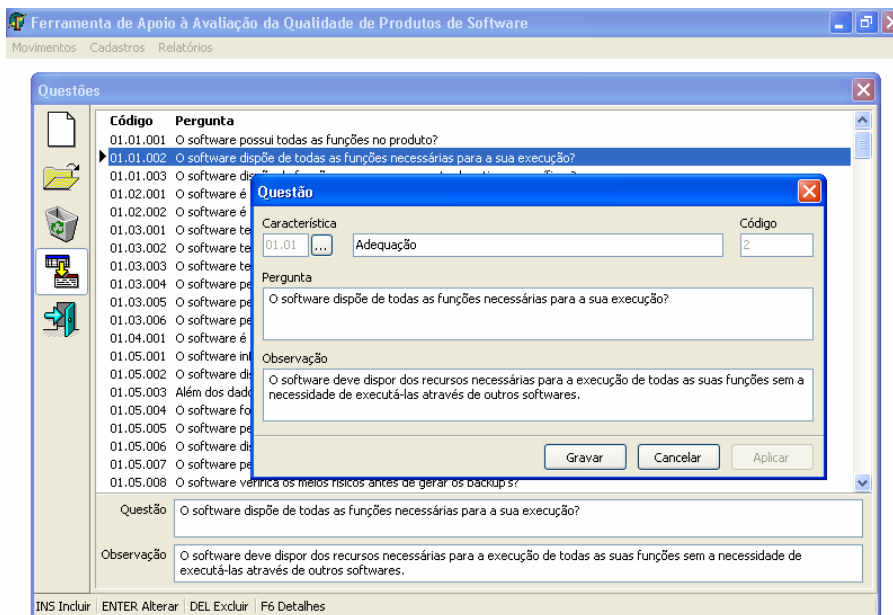


Excluído: 4

Excluído: 1

Excluído: 17

Figura 4-10 – Tela de cadastro das questões



Excluído: 4

Excluído: 1

Excluído: 19

- f) cadastro de questões: na [Figura 4-10](#) está representado o cadastro de questões em que o usuário/avaliador deverá entrar com os dados das questões associando cada uma delas à uma subcaracterística da qualidade. As questões serão visualizadas no momento das avaliações. Ainda no cadastro da questão é possível informar uma observação para a mesma no intuito de ajudar quem está avaliando o software. Nesta tela também são executadas as manutenções do cadastro de questões;
- g) tipos de avaliação: após a entrada de todos os dados que farão parte da avaliação, o usuário/avaliador inicia o processo de configurar a avaliação, cadastrando os tipos de avaliação conforme a [Figura 4-11](#). Nesta opção o usuário/avaliador poderá definir padrões de avaliação como, por exemplo, selecionar somente as características e subcaracterísticas mais relevantes para avaliar um software financeiro. Nesta tela também são executadas as manutenções do cadastro de tipos de avaliação;

Excluído: Figura 4-19

Excluído: Figura 1-19

Excluído: Figura 4-20

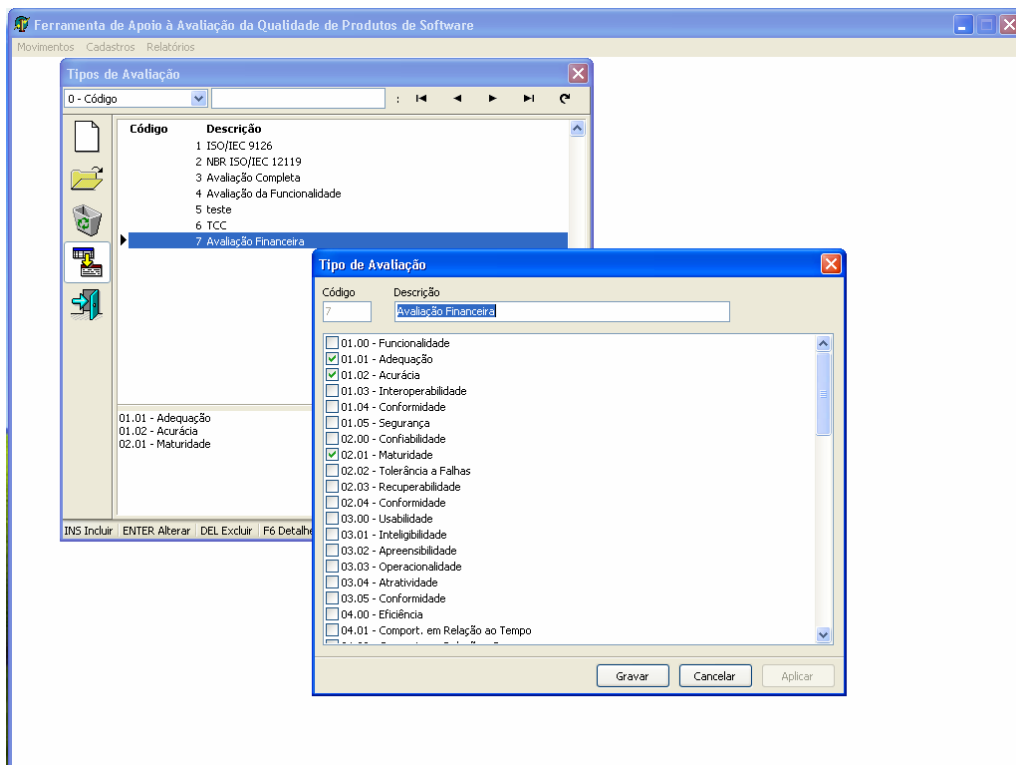
Excluído: Figura 1-20

Excluído: 4

Excluído: 1

Excluído: 20

**Figura 4-11 – Tela dos tipos de avaliação**





- h) avaliação: após a entrada de todos os dados que farão parte da avaliação, incluindo os tipos de avaliação o usuário/avaliador inicia o processo da avaliação propriamente dita. Na tela inicial, após clicar no botão incluir, o usuário/avaliador poderá trabalhar com a tela inicial da avaliação, conforme mostra a [Figura 4-12](#), onde deverá informar alguns dados para começar a avaliação, tais como: o objetivo da avaliação, o avaliador que fará a avaliação, o software a ser avaliado, alguns dados do hardware onde será feita a avaliação, alguns dados em relação ao software a ser avaliado, o tipo de avaliação que será feito neste software e uma observação sobre a avaliação, se necessário. Nesta tela também são executadas as manutenções das avaliações;

Excluído: Figura 4-22

Excluído: Figura 1-22

**Figura 4-12 – Tela inicial da avaliação**

Excluído: 4

Excluído: 1

Excluído: 22

Código	Objetivo	Avaliador	Software	Espaço em Disco	Hardware Especial	Hardware	Impressora	Manual
1	Financeiro - ISO/IEC 9126	1	1	16GB	SCANNER	DURON 1,2GHZ	HP 656C	MANUAL DC
2	Financeiro 1 - NBR ISO/IEC 121	1	1	16GB	SCANNER	DURON 1,2GHZ	HP 656C	DO USUÁRIO
3	Financeiro 3 - NBR ISO/IEC 121	2	3	16GB	SCANNER	DURON 1,2GHZ	HP 656C	ON-LINE
4	Financeiro 4	1	1					
5	Financeiro 5	1	1					
6	Avaliação Financeira	1	3	16GB	SCANNER	DURON 1,2GB	HP 656C	MANUAL DC
7	Segunda Avaliação Financeira	1	3	16GB	SCANNER	DURON 1,2GB	HP 656C	MANUAL DC

**Avaliação**

Código: 7    Objetivo: Segunda Avaliação Financeira    Período: Data Inicial: 10/ 7 /2003    Data Final: 10/ 7 /2003

Avaliador: 1    Fabiano Stinamiglio Ferreira    Hora Inicial: 21:00    Hora Final: 22:00

Software: 3    Financeiro

**Dados do Hardware**

Hardware: DURON 1,2GB  
Hardware Especial: SCANNER  
Espaço em Disco: 16GB  
Impressora: HP 656C  
Monitor: AOC 15"  
Mídia: CD-ROM  
Outros: TECLADO SEM-FIO

**Outros Dados**

Software Especial: INTERNET EXPLORER  
Manual: MANUAL DO USUÁRIO  
RAM: 256 MB  
S.O.: WINDOWS XP

**Observações**

AVALIAÇÃO P/ AQUISIÇÃO DE UM SOFTWARE FINANCEIRO DE ACORDO COM A NORMA ISO/IEC 9126

Tipo de Avaliação: 7    Avaliação Financeira

Avançar    Gravar    Cancelar    Aplicar

AVALIAÇÃO P/ AQUISIÇÃO DE UM SOFTWARE FINANCEIRO DE ACORDO COM A NORMA ISO/IEC 9126

INS Incluir    ENTER Alterar    DEL Excluir    F6 Detalhes    F11 Ordem

- i) respondendo a avaliação: após preencher a tela inicial da avaliação, o usuário/avaliador clica no botão avançar para começar a responder a avaliação, conforme a [Figura 4-13](#). Esta tela mostrará as características e subcaracterísticas da

Excluído: Figura 4-24

Excluído: Figura 1-24

qualidade com seus pesos respectivos que poderão ser alterados pelo usuário/avaliador durante a avaliação. Para cada características/subcaracterísticas, serão mostradas as questões e observações referente a cada uma delas, onde o usuário/avaliador deverá informar a resposta à questão, clicando numa das opções disponíveis. Após responder a uma questão, permite-se ir para a próxima questão, voltar à questão anterior ou ainda voltar para a tela inicial da avaliação, mas é importante ressaltar que o botão de gravação da avaliação só fica disponível nas telas de movimentações, para evitar gravar uma avaliação em branco, ou seja, pelo menos uma questão deve ser respondida para se poder gravar a avaliação. Esta tela pode ser visualizada na [Figura 4-13](#). Ainda nesta tela há o botão calcular resultado, que durante a avaliação, o usuário/avaliador poderá visualizar o resultado parcial da avaliação, conforme mostra a [Figura 4-14](#).

Excluído: Figura 4-24

Excluído: Figura 1-24

Excluído: Figura 4-26

Excluído: Figura 1-26

Excluído: 4

Excluído: 1

Excluído: 24

**Figura 4-13 – Movimentação da avaliação**

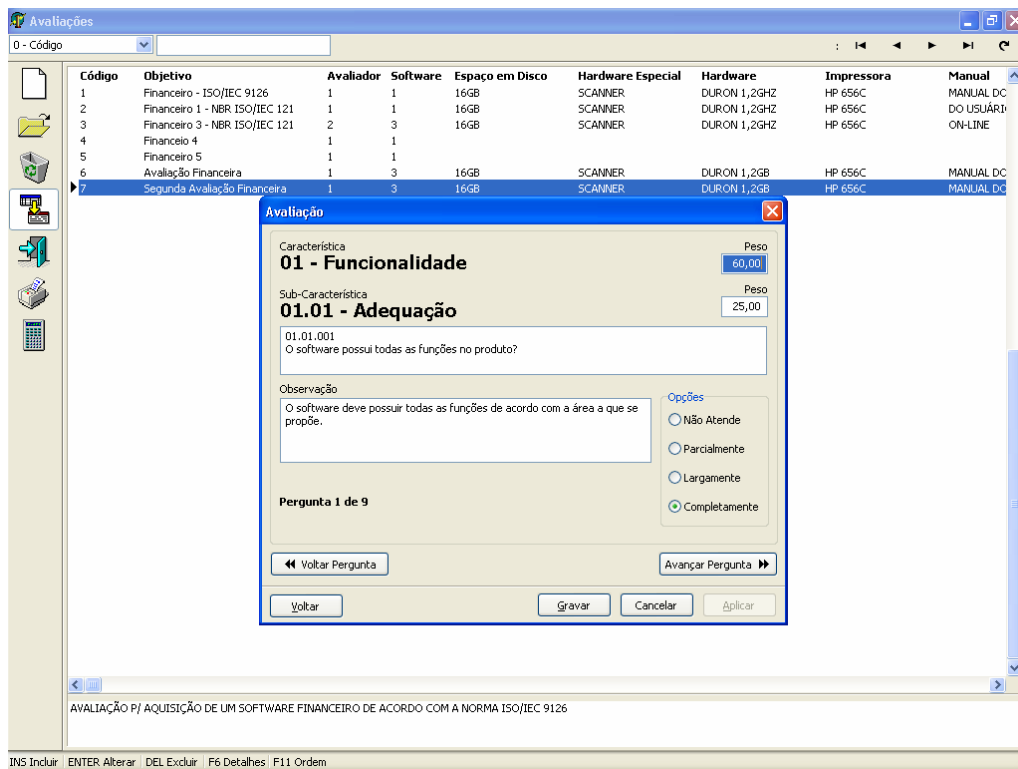


Figura 4-14 – Tela de resultado da avaliação

Característica	Descrição	Peso	Resultado	Percentual
01.00	<b>Funcionalidade</b>	60.00	47.7273	79.55 %
01.01	Adequação	25.00	25.0000	100.00 %
01.02	Acurácia	30.00	18.7500	62.50 %
02.00	<b>Confiabilidade</b>	40.00	30.0000	75.00 %
02.01	Maturidade	15.00	11.2500	75.00 %

Resultado Geral: **77.73 %**      Classificação: **Bom**      Aceito

Excluído: 4

Excluído: 1

Excluído: 26

- j) relatório de avaliação do produto de software: no Apêndice B, pode ser verificado o primeiro relatório do protótipo, chamado de relatório da avaliação e trata-se de um espelho da avaliação, mostrando de forma estruturada o que aconteceu naquela avaliação. Traz ordenado por característica com o peso respectivo, após as subcaracterísticas com os pesos respectivos e as questões que foram respondidas com suas respostas. O relatório também mostra todos os dados informados na tela inicial da avaliação, o resultado de cada característica e qual foi a sua classificação final;
- k) relatório comparativo: no Apêndice C, pode ser verificado o relatório que lista duas avaliações já executadas para fins de comparar qual produto possui mais qualidade. Os dados deste relatório são os mesmos citados no item anterior, ou seja, possui os mesmos dados que o relatório da avaliação;
- l) avaliação em branco: no Apêndice D, pode ser verificado o relatório que lista uma avaliação em branco, para que possa ser preenchida pelo usuário/avaliador. Neste relatório o usuário/avaliador pode escolher qual tipo de avaliação deseja imprimir, flexibilizando a avaliação.

## 4.4 CONSIDERAÇÕES SOBRE A IMPLEMENTAÇÃO

Para a implementação foram utilizados alguns componentes da linguagem Delphi 5.0, componentes para tratamento da interface, impressão de dados, leitura e gravação. Por tratar-se de uma base de dados padrão, utilizou-se para gravação de arquivos, tabelas Paradox 7.0.

Utilizou-se para os menus, forms com labels. Para o item de menu Cadastros, foram utilizados Panel para agrupamento de componentes de entrada de dados. Os componentes utilizados Edit, MaskEdit, DBGrid para a entrada dos dados e Labels e GroupBox para trazer a descrição dos respectivos campos. Usou-se também o DataModule no qual estão os componentes relativos às tabelas (Table).

No desenvolvimento da interface da ferramenta, utilizou-se o componente Theme Manager que, além de dar à ferramenta características estruturais de imagem semelhantes ao Windows XP, torna a visualização da ferramenta mais agradável, dando suavidade em todas as formas e campos de uma tela (o componente Theme Manager só terá funcionalidade quando executado no Windows XP).

Para desenvolvimento dos relatórios utilizou-se QuickReport 3.0, que acompanha a instalação do Delphi 5.0. Optou-se pelo seu uso devido à diversidade de funções que o mesmo disponibiliza em tempo de desenvolvimento e de execução, pela facilidade de manipulação do layout dos campos de um relatório.

## 5 CONCLUSÕES

Este trabalho procurou demonstrar a aplicação das normas ISO/IEC 9126 e NBR ISO/IEC 12119 que tratam da qualidade de produtos de software.

A utilização destas normas torna o processo de avaliação de software mais simples e prático, pois designam etapas a serem cumpridas e disponibilizam ao avaliador características eficazes para a avaliação da qualidade de produtos de software.

Quanto à ferramenta, pode-se afirmar que é de fácil operação, permitindo que pessoas leigas possam utilizá-la na realização da avaliação de diversos produtos de software. Além disso, uma das grandes vantagens da utilização da ferramenta é a criação de base histórica, permitindo assim ao avaliador comparar avaliações anteriores com futuras avaliações, podendo também comparar dois ou mais produtos de software de mesma categoria, para saber qual a melhor opção na escolha de um produto de software. Outra grande vantagem da sua utilização é a flexibilidade para poder configurar a avaliação com as características da qualidade que melhor lhe convir, deixando gravada esta configuração e podendo utilizá-la na avaliação que desejar.

Pode-se dizer que os objetivos deste trabalho foram alcançados, visto que, com base em trabalhos de conclusão de curso já realizados na área de qualidade de software, entrevistas realizadas com profissionais da área (consultores de software e validadores de software) e também com usuários de software, foi desenvolvido um questionário de avaliação de produtos de software contendo as principais características e subcaracterísticas da qualidade de acordo com as normas ISO/IEC 9126 e NBR ISO-IEC 12119, e que pode ser utilizado na avaliação de softwares de diversas categorias.

A grande diferença da ferramenta desenvolvida neste trabalho para as demais ferramentas desenvolvidas em trabalhos correlatos citados anteriormente está na abrangência das Normas ISO/IEC 9126 e NBR ISO/IEC 12119 onde, em todos os trabalhos correlatos já citados, procurou-se avaliar produtos de software de acordo com somente uma das Normas e, na sua maioria, estão voltados somente para a avaliação de um tipo ou categoria de software específico. O avanço que houve neste trabalho foi justamente a união dos trabalhos correlatos já existentes, filtrando o que há de melhor em cada um deles, incrementando ainda mais o

*checklist* através de entrevistas e pesquisas descritas neste trabalho, inovando a ferramenta, como por exemplo, poder escolher determinadas subcaracterísticas para serem avaliadas, não necessitando escolher toda uma característica, e também permitir a emissão de relatórios comparativos entre duas avaliações, visualizando melhor as diferenças entre uma avaliação e outra, além de permitir a avaliação de qualquer tipo ou categoria de produto de software.

Ainda em relação ao *checklist*, é importante ressaltar que não foram desenvolvidas questões referente o item “Instruções para Teste” da Norma NBR ISO/IEC 12119 pelo fato de que este item especifica como um produto deve ser testado em relação aos requisitos de qualidade, ou seja, após já possuir um questionário para avaliação de produtos de software, o item Instruções para Teste serve como um guia de como aplicar este questionário na avaliação de produtos de software. Por este motivo não foram elaboradas questões sobre este item, mas o mesmo foi utilizado como um guia de teste durante a avaliação dos softwares feitos pela ferramenta desenvolvida.

Por fim, fica claro que uma ferramenta de classificação de software de diversas categorias não é amplamente aceita no mercado devido a dificuldade de inserir cada categoria de software e suas informações específicas as quais necessitam ser avaliadas e com grande relevância, o que uma ferramenta genérica como a proposta neste trabalho não atende completamente. Mas o intuito desta ferramenta é verificar os requisitos mínimos de qualidade que um produto de software deve possuir para ser utilizado, pois sem estes requisitos mínimos, será difícil alguém querer utilizá-los.

## 5.1 SUGESTÕES

Para trabalhos futuros e aprimoramento do mesmo sugere-se:

- a) ampliar o detalhamento das características e subcaracterísticas da qualidade das normas ISO/IEC 9126 e NBR ISO/IEC 12119 permitindo uma avaliação mais completa;
- b) aplicar mais testes em produtos de software de diferentes categorias e com diferentes avaliadores para comprovar a utilizabilidade e funcionalidade da ferramenta;
- c) tornar a ferramenta atual mais amigável, disponibilizando telas de consultas e relatórios cadastrais;

- d) tornar a ferramenta atual mais flexível em relação às características da qualidade, fazendo com que seja possível cadastrar mais níveis de subcaracterísticas, fazendo com que a avaliação se torne mais minuciosa;
- e) Implementar à ferramenta a utilização de métricas de software mundialmente conhecidas, para tornar a ferramenta ainda mais consistente e efetiva.

## REFERÊNCIAS BIBLIOGRÁFICAS

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 12119**: tecnologia de informação – pacotes de software – teste e requisitos de qualidade. Rio de Janeiro, 1998.

ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR 13596**: tecnologia de informação – avaliação de produto de software – características de qualidade e diretrizes para o seu uso. Rio de Janeiro, 1996.

ANTONIONI, J. A. Rosa. **Qualidade em software**: manual de aplicação da ISO-9000. São Paulo: Makron Books, 1995.

BARRETO JUNIOR, José. **Qualidade de pacotes de software – ISO 12119**. Disponível em: <<http://www.inf.ufsm.br/~oliveira/elc311/qualidadeSW.html#Pacotes>>. Acesso em: 20 jun. 2002.

BOEGH, J. et al. **A practitioners guide to evaluation of software**. Brighton, Software Engineering Standards Symposium, 1983.

BOEGH, J. et al. **A method for software quality planning, control and evaluation**. IEEE Software, 1999.

CAMPOS, Vicente Falconi. **Controle da qualidade total**. Rio de Janeiro: Fundação Christiano Ottoni, 1994.

CÂNDIDA, Inthum. **Qualidade & teste de software**: engenharia de software, qualidade de software, qualidade de produtos de software, teste de software, formalização do processo de teste, aplicação prática dos testes. Florianópolis: Visual Books, 2001.

CANTU, Marco. **Dominando o Delphi 5**: a bíblia. São Paulo: Makron Books, 2000.

FERNANDES, Aguinaldo Aragon. **Gerência de software através de métricas**. São Paulo: Atlas, 1995.

FISHER, Alan S. **CASE**: utilização de ferramentas para desenvolvimento de software. Rio de Janeiro: Campus, 1990.



GUESSER, Enezilda. **Protótipo de sistema para auxiliar a avaliação da qualidade de sistemas de automação comercial**. 1995. 116 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

ISO/IEC 9126. **Tecnologia de informação**. Modelo de qualidade para produto de software, Rio de Janeiro, jan. 2000. Disponível em: <[http://www.abnt.org.br/index\\_body.htm](http://www.abnt.org.br/index_body.htm)>. Acesso em: 01 nov. 2002.

JUNCKES, Fabio Alexandre. **Software de apoio ao processo de auditoria segundo normas de qualidade**. 1999. 66 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

KOSCIANSKI, André et al. **Guia para utilização de normas sobre qualidade de produto de software ISO/IEC 9126 e ISO/IEC 14598**. ABNT – Associação brasileira de normas técnicas. SC10 – subcomitê de software. Curitiba, 1999.

MARTIN, James; MCCLURE, Carma. **Técnicas estruturadas e CASE**. São Paulo: Makron, McGraw-Hill, 1991.

NÓBREGA, Kleber Cavalcanti. **Apostila do curso de gestão da qualidade - especialização em gestão da qualidade**. Blumenau, 1994.

PISKE, Rosilene. **Protótipo de sistema para avaliação da qualidade de softwares de folha de pagamento**. 1996. 92 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

POFFO, Marcio Alberto. **Protótipo para a avaliação da qualidade de softwares de contabilidade**. 1995. 90 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

SANDERS, Joc; CURRAN, Eugene. **Qualidade de software**. Disponível em: <<http://www.geekbrasil.com.br>>. Acesso em: 12 jan. 2003.

SANDRI, Vivian. **Software de apoio a avaliação da qualidade de pacotes baseado na norma ISO/IEC 12119**. 1998. 60 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

STORCH, Mirian Mirdes. **Proposta de avaliação da qualidade de produtos de software utilizando a norma ISO/IEC 9126**. 2000. 82 f. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) – Centro de Ciências Exatas e Naturais, Universidade Regional de Blumenau, Blumenau.

WEBER, Kival Chaves; ROCHA, Ana Regina Cavalcanti da; NASCIMENTO, Célia Joseli do. **Qualidade e produtividade em software**. São Paulo: Makron Books, 2001.

# **APÊNDICE A – LISTAGEM DO *CHECKLIST* DESENVOLVIDO A PARTIR DAS NORMAS ISO/IEC 9126 E NBR ISO/IEC 12119**

## **1. FUNCIONALIDADE**

### **1.1 Adequação**

- 1.1.1 O software possui todas as funções 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 multi-usuário?
- 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?
- 1.3.4 O software permite exportar os dados armazenados?
- 1.3.5 O software permite importar dados de outros softwares?
- 1.3.6 O software permite integração on-line com outros softwares?

### **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 backup's na instalação de novas versões?

- 1.5.2 O software dispõe de rotina interna de backup?
- 1.5.3 Além dos dados, é possível fazer backup do software em si?
- 1.5.4 O software força o backup periodicamente?
- 1.5.5 O software permite a compactação de backup's?
- 1.5.6 O software dispõe de rotina interna de restore?
- 1.5.7 O software permite executar as rotinas de backup e restore para/de meios físicos variados?
- 1.5.8 O software verifica os meios físicos antes de gerar os backup's?
- 1.5.9 O software possui controle de versão na restauração?
- 1.5.10 O acesso ao software é protegido com senhas, contra usuários não autorizados?
- 1.5.11 O software permite estabelecer uma política de controle de acesso em nível de funções/rotinas?
- 1.5.12 O software permite estabelecer uma política de controle de acesso em nível de arquivo?
- 1.5.13 É possível alterar, externamente, os dados dos arquivos?
- 1.5.14 Os arquivos de dados são criptografados?

## **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 os dados na queda de energia durante o processamento de atualização de dados?

### **2.2 Tolerância às falhas**

- 2.2.1 O software possui advertência de erros cometidos pelo usuário?
- 2.2.2 O software possui advertência quando ocorrem erros na recuperação de arquivos?
- 2.2.3 O software garante advertência quando ocorrem 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 faz consistência dos campos controlando dados que fogem da média?
- 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?
- 2.2.16 Há alarme sonoro quando da detecção de erros?

### **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 casos de erros de execução ocorrem erros de informação?

- 2.3.5 O software tem capacidade de recuperar dados excluídos?
- 2.3.6 A rotina de recuperação do software é fácil de ser usada?
- 2.3.7 A recuperação dos dados ocorre de forma rápida?
- 2.3.8 O software possui uma função de reversão de uma operação errônea?

## **2.4 Conformidade**

- 2.4.1 O software é conciso nos resultados, passando confiança ao usuário?

## **3. USABILIDADE**

### **3.1 Inteligibilidade**

- 3.1.1 O software possui autodemonastração?
- 3.1.2 A autodemonastração permite compreender como o software funciona?
- 3.1.3 A autodemonastração é facilmente acessada?
- 3.1.4 O software possui tutorial on-line?
- 3.1.5 Após a instalação, o software oferece a possibilidade de usar o tutorial?
- 3.1.6 As telas do software são auto-instrutivas, permitindo ao usuário visualizar com facilidade qual sua função?
- 3.1.7 As telas do mesmo nível possuem 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 mantém 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 do software 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 nível de detalhamento do índice permite a fácil localização da informação desejada?
- 3.2.9 O manual de operação do software inclui os termos técnicos utilizados?
- 3.2.10 Existe uma hierarquia de manuais de acordo com o nível de conhecimento do usuário?
- 3.2.11 Os termos são usados com o mesmo significado durante todo o processamento?
- 3.2.12 Todas as figuras do software estão explicadas no manual?
- 3.2.13 Todos os passos para a instalação do software estão claramente apresentados?
- 3.2.14 As configurações mínimas de hardware para a instalação estão claramente definidas?
- 3.2.15 Os possíveis erros de instalação são apresentados claramente no manual?
- 3.2.16 O manual apresenta exemplo de como utilizar o software?
- 3.2.17 O manual apresenta todos os erros que podem ocorrer no software, apresentando o erro ocorrido, as possíveis causas e as possíveis soluções?
- 3.2.18 Os textos dos manuais são corretamente escritos (sem erros sintáticos e semânticos)?
- 3.2.19 O manual do usuário explica as convenções de estilo utilizadas durante o documento (negrito, itálico, etc.)?
- 3.2.20 O volume de texto está de acordo com a quantidade de informações obtidas?

- 3.2.21 O layout do texto no manual possibilita uma leitura agradável?
- 3.2.22 O software possui ajuda on-line?
- 3.2.23 O software possui ajuda on-line de como corrigir os erros cometidos pelo usuário?
- 3.2.24 A ajuda on-line possui um índice?
- 3.2.25 A ajuda on-line possui busca por palavra chave?
- 3.2.26 O nível de detalhamento do índice permite a rápida localização da informação desejada?
- 3.2.27 Os textos da ajuda on-line utilizam recursos de hipertexto?
- 3.2.28 Os termos utilizados possuem o mesmo significado em todo arquivo de ajuda?
- 3.2.29 A ajuda é sensível ao contexto?
- 3.2.30 As informações contidas em cada seção da ajuda estão de acordo com o título desta seção?
- 3.2.31 As mensagens de ajuda apresentam uma explicação para todos os itens relacionados à utilização do software?
- 3.2.32 A linguagem utilizada na mensagem de ajuda é facilmente entendida pelo usuário?
- 3.2.33 Os termos técnicos (específicos ou de informática) utilizados nas mensagens de ajuda são claramente explicados ao usuário?
- 3.2.34 O software mostra como o usuário deve navegar pelo arquivo de ajuda?
- 3.2.35 Pode-se navegar pela ajuda através do teclado e do mouse?
- 3.2.36 Ao acessar a tela de ajuda o usuário é informado sobre o número de telas que deverá ler para ter uma compreensão do item solicitado?
- 3.2.37 As mensagens/formatos/layout's/apresentações de orientações estão padronizadas?

### **3.3 Operacionalidade**

- 3.3.1 A interface do software é amigável?
- 3.3.2 O software pode ser modificado de forma a ser personalizado por usuário?
- 3.3.3 Os comandos utilizados tem a mesma finalidade em todas as funções do software?



- 3.3.4 Os comandos do software estão de acordo com os padrões existentes (F1, DEL, ESC, ENTER)?
- 3.3.5 É possível prever o que existe dentro de cada opção do menu?
- 3.3.6 A disposição dos menus é adequada à utilização do software?
- 3.3.7 Existe padronização de teclas de função?
- 3.3.8 O software possui atalhos (combinações de teclas) para os usuários mais experientes?
- 3.3.9 O software possui uma lista de atalhos fácil de ser acessada?
- 3.3.10 As apresentações dos resultados através de tabelas ou relatórios são suficientes para uma boa utilização do software?
- 3.3.11 O software permite executar processos em background?
- 3.3.12 Durante o processo de alguma operação mais demorada, o software apresenta alguma informação do que está ocorrendo?
- 3.3.13 É informado ao usuário qual é a sua posição no software?
- 3.3.14 O usuário tem a possibilidade de parar algum processamento demorado?
- 3.3.15 O software possui mensagens de alerta claros para as consequências de uma determinada confirmação?
- 3.3.16 As opções do menu podem ser acessadas por letras ou combinações de teclas?
- 3.3.17 As letras utilizadas para escolher as opções do menu são adequadas?
- 3.3.18 O software possui teclas de função para acesso a funções específicas?
- 3.3.19 As janelas do software podem mudadas de posição?
- 3.3.20 É possível acessar outro menu ou opção sem abandonar a atual?
- 3.3.21 O software permite ao usuário alterar cores?
- 3.3.22 O software permite a modificação de cores das janelas de todos os níveis?
- 3.3.23 O software impede a combinação de cores que torna impossível a visualização (fundo branco com letras brancas, por exemplo)?

3.3.24 A rotina de configuração de cores é fácil de ser utilizada?

3.3.25 Há homogeneidade de formato entre letras?

3.3.26 Ao terminar uma tarefa o software informa de maneira adequada ao usuário sobre a conclusão da mesma?

3.3.27 Em caso de alarme sonoro, o operador pode ativar/desativar o alarme?

3.3.28 O software permite ao usuário eliminar campos?

3.3.29 O software permite ao usuário acrescentar campos?

3.3.30 O software permite ao usuário tornar um campo opcional ou obrigatório?

### **3.4 Atratividade**

3.4.1 O software possui recursos atrativos para o usuário?

### **3.5 Conformidade**

3.5.1 O software possui padrões de utilização, não deixando que o usuário fique perdido se aberto várias telas, por exemplo?

## **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 ao volume de dados envolvido?

4.1.5 O tempo de resposta é adequado à 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 backup's é satisfatório?

### **4.2 Utilização dos recursos**

4.2.1 Os recursos do equipamento exigidos pelo software são adequados à 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 de memória?

4.2.4 O software faz muito acesso a disco?

4.2.5 O software permite a sua operação durante a impressão de documentos?

### **4.3 Conformidade**

4.3.1 O software executa os procedimentos num tempo satisfatório?

## **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 as decisões comentadas?

5.1.7 O software possui variáveis que são descritas por comentários?

5.1.8 O software possui os desvios comentados?

5.1.9 O software possui toda a programação em linguagem de máquina comentada?

5.1.10 O tamanho de uma unidade não é superior a uma página?

5.1.11 Os nomes de todas as variáveis do software são exclusivas?

5.1.12 As variáveis são usadas apenas de uma única forma?

5.1.13 As variáveis globais são usadas consistentemente em relação às unidades 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 O software verifica possíveis conflitos ou combinações ilegais 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?

5.2.8 O software permite ser alterado quando da mudança de legislação?

## **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?

## **5.5 Conformidade**

- 5.5.1 As manutenções no software são registradas?
- 5.5.2 O software possui um processo de manutenção padrão, ou seja, todos seguem um mesmo procedimento para fazer manutenção no software?

## **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 se 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 ajuda na instalação?
- 6.2.4 O software possui uma ordem lógica na seqüência dos disquetes ou cd's 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 backup antes de fazer a instalação da nova versão?
- 6.2.10 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 auto-sugestivo nas opções de instalações?
- 6.2.13 É possível definir a unidade e o local onde se deseja instalar o software?
- 6.2.14 É avisado ao usuário sobre a necessidade de dar um reboot na máquina após a instalação?
- 6.2.15 É criado o grupo de programa quando for o caso?
- 6.2.16 São criados os itens de programa quando for o caso?
- 6.2.17 São criado os ícones de para acesso e execução do software quando for o caso?
- 6.2.18 É fornecido uma função que permite interromper o processo de instalação?
- 6.2.19 É criado o grupo de programa quando for o caso?

### **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 executa todas as funções que deveria executar?
- 6.3.4 O software se aplica a outros fins, além do projetado?
- 6.3.5 O software se adequa ao fim a que se destina?
- 6.3.6 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?

## **6.5 Coexistência**

6.5.1 O software tem capacidade de coexistir?

## **7. REQUISITOS PARA DESCRIÇÃO DE PRODUTOS**

### **7.1 Requisitos Gerais sobre o Conteúdo da Descrição de Produto**

7.1.1 O Produto possui uma Descrição do Produto?

7.1.2 A Descrição do Produto é suficientemente compreensível?

7.1.3 É fácil ter uma visão geral pela Descrição do Produto?

7.1.4 A Descrição de Produto está livre de inconsistências internas?

7.1.5 O significado dos termos utilizados permanece constante em toda a Descrição de Produto?

### **7.2 Identificações e Indicações**

7.2.1 O Produto possui uma identificação da Descrição do Produto?

7.2.2 Descrição do Produto possui uma identificação do produto?

7.2.3 Se existem variações do produto (como versão Cliente/Servidor ou Profissional), cada variação apresenta o nome do produto, nome da variante e uma versão ou data?

7.2.4 Descrição do Produto possui uma identificação do produtor?

7.2.5 A Descrição do Produto possui uma identificação do fornecedor?

7.2.6 Através da Descrição do Produto, é possível identificar facilmente a função principal do produto?

7.2.7 Alguns produtos de software devem atender a certos requisitos estabelecidos por lei, norma ou convenção (Ex.: Folha de Pagamento). Se o produto enquadra-se nesta situação, sua descrição faz referência a todos os documentos com os quais está em conformidade?

7.2.8 A Descrição do Produto especifica requisitos mínimos recomendados de Hardware e Software?

### **7.3 Funcionalidade**

7.3.1 A Descrição do Produto fornece uma visão geral das funções disponíveis para o usuário, dos dados necessários e das facilidades oferecidas?

7.3.2 Caso o uso do produto seja limitado por valores específicos, estes são fornecidos?

7.3.3 Caso existam mecanismos de segurança para impedir acesso não autorizado (acidental ou intencional), estes recursos são declarados?

### **7.4 Confiabilidade**

7.4.1 Existem informações sobre procedimentos para a realização de cópia de segurança de dados (backup)?

### **7.5 Usabilidade**

7.5.1 É declarado qual o tipo de interface utilizada para operar o programa?

7.5.2 É declarada a existência de um sistema de ajuda (help)?

7.5.3 É declarado algum conhecimento específico para o uso da aplicação?

7.5.4 Caso o produto utilize na documentação e/ou interface, outros idiomas diferentes do utilizado na Descrição do Produto, estes são declarados?

7.5.5 Se o produto pode ser adaptado pelo usuário, estas possibilidades estão descritas?

7.5.6 Se o produto possui algum tipo de proteção contra infrações a direitos autorais que dificulte seu uso, esta proteção é declarada?

7.5.7 A Descrição do Produto inclui dados sobre eficiência de uso e satisfação do usuário?

### **7.6 Eficiência**

7.6.1 A Descrição do Produto inclui dados sobre tempo de resposta, desempenho ou similares (com base em uma configuração específica)?

### **7.7 Manutenibilidade**

7.7.1 A Descrição do Produto contém declarações sobre manutenibilidade?



## **7.8 Portabilidade**

7.8.1 A Descrição do Produto contém declarações sobre Portabilidade?

## **8. REQUISITOS PARA DOCUMENTAÇÃO DO USUÁRIO**

### **8.1 Completitude**

8.1.1 Todas as funções apresentadas na Descrição do Produto estão completamente descritas na Documentação do Usuário?

8.1.2 Todas as funções do produto estão completamente descritas na Documentação do Usuário?

8.1.3 Todos os valores-limite apresentados na Descrição do Produto estão na Documentação do Usuário?

8.1.4 Caso a instalação possa ser feita pelo usuário, a Documentação do Usuário inclui um manual contendo todas as informações necessárias para a realização desta tarefa?

8.1.5 Caso a manutenção do produto possa ser feita pelo usuário, a Documentação do Usuário inclui um manual contendo todas as informações necessárias para a realização desta tarefa?

### **8.2 Corretude**

8.2.1 Todas as informações contidas na Documentação do Usuário estão corretas?

8.2.2 A Documentação do Usuário está livre de ambigüidades?

### **8.3 Consistência**

8.3.1 A Documentação do Usuário está livre de contradições internas (dentro de um documento único ou entre documentos)?

8.3.2 A Documentação do Usuário está livre de contradições com a Descrição do Produto?

8.3.3 O significado dos termos utilizados permanece constante em toda a Documentação do Usuário?

### **8.4 Inteligibilidade**

8.4.1 A Documentação do Usuário é compreensível pelos usuários típicos do produto?

### **8.5 Facilidade de Visão Geral**

8.5.1 A Documentação do Usuário é bem organizada?

8.5.2 A Documentação do Usuário possui boa apresentação?

8.5.3 É fácil ter uma visão geral da Documentação do Usuário?

8.5.4 Todos os documentos possuem Índice Remissivo?

8.5.5 Todos os documentos possuem Índice Analítico?

8.5.6 Se a Documentação do Usuário não é fornecida no formato impresso, há instruções sobre como imprimi-la?

## **9. REQUISITOS PARA PROGRAMAS E DADOS**

### **9.1 Funcionalidade**

9.1.1 Caso a instalação possa ser realizada pelo usuário, as informações contidas no manual de instalação são suficientes para realizar esta tarefa?

9.1.2 Os requisitos de hardware e software apresentados na Descrição do Produto são suficientes para realizar a instalação?

9.1.3 Após a instalação, é possível verificar (através de casos de testes previamente fornecidos ou através de auto-testes com mensagens de status) se o software funciona?

9.1.4 Todas as funções mencionadas na Documentação do Usuário são executadas na forma ali descritas com as facilidades, propriedades e dados correspondentes e dentro dos valores-limite fornecidos?

9.1.5 Os programas e dados existentes correspondem às declarações apresentadas na Descrição do Produto e na Documentação do Usuário?

9.1.6 As funções são executadas de uma maneira correta para as tarefas a que se propõem?

9.1.7 Os programas e dados estão livres de contradições internas?

9.1.8 Os programas e dados estão livres de contradições com a Descrição do Produto?

- 9.1.9 Os programas e dados estão livres de contradições com a Documentação do Usuário?
- 9.1.10 Todos os termos utilizados possuem o mesmo significado em todo o software?
- 9.1.11 O controle da operação do programa (formato de mensagens, telas, relatórios, etc.) e o seu comportamento seguem um padrão?

## **9.2 Confiabilidade**

- 9.2.1 Quando a capacidade do sistema (link) é explorada até os limites especificados, o sistema apresenta-se confiável?
- 9.2.2 Quando da tentativa de explorar a capacidade do sistema acima dos limites especificados, o sistema apresenta-se confiável?
- 9.2.3 Ao receber uma entrada incorreta proveniente do usuário ou de outros programas citados na Descrição do Produto, o sistema apresenta-se confiável?
- 9.2.4 O sistema recusa entradas de dados sintaticamente incorretas? Ao serem violadas instruções explícitas contidas na Documentação do Usuário, o sistema apresenta-se confiável?

## **9.3 Usabilidade**

- 9.3.1 As perguntas, mensagens e resultados dos programas são compreensíveis?
- 9.3.2 As mensagens de erro oferecem informações detalhadas explicando a causa de um erro?
- 9.3.3 As mensagens de erro oferecem informações detalhadas explicando o que deve ser feito para corrigir um erro?
- 9.3.4 Cada meio de armazenamento de dados (mídia) contém uma identificação do produto?
- 9.3.5 É possível o usuário identificar a função que está sendo executada?
- 9.3.6 As informações fornecidas ao usuário são de fácil visualização e leitura?
- 9.3.7 As informações dos programas estão agrupadas (distribuição das funções pelos itens de menu, localização dos dados dentro das janelas, etc.) e codificadas (termos utilizados para descrição das funções e dados) de forma apropriada?

9.3.8 As mensagens fornecidas pelos programas são projetadas para que o usuário as diferencie facilmente por tipo?

9.3.9 Os formatos das telas de entrada, relatórios e outras entradas e saídas são claros e de fácil visualização?

#### **9.4 Operacionalidade**

9.4.1 A execução de funções com sérias conseqüências são reversíveis?

9.4.2 Ao executar tais funções, é dado um claro aviso de suas conseqüências e é solicitada uma confirmação antes de executá-las?

9.4.3 Se a documentação é disponibilizada de forma on-line, o usuário pode acessar subtópicos de um assunto de uma maneira direta?

#### **9.5 Eficiência**

9.5.1 Caso existam declarações sobre eficiência na Descrição do Produto, elas estão em conformidade com o produto?

#### **9.6 Manutenibilidade**

9.6.1 Caso existam declarações sobre manutenibilidade na Descrição do Produto, elas estão em conformidade com o produto?

#### **9.7 Portabilidade**

9.7.1 Caso existam declarações sobre portabilidade na Descrição do Produto, elas estão em conformidade com o produto?

## APÊNDICE B – RELATÓRIO DE AVALIAÇÃO DO PRODUTO DE SOFTWARE

Impressão da Avaliação

Identificação da Avaliação: 000007 Segunda Avaliação Financeira

Avaliador: 1 - Fabiano Stiramiglio Ferreira  
 Produtor: 3 - Software House 3  
 Produto: 3 - Financeiro Versão: 2.0

Informações do Produto  
 Sistema Financeiro

Materiais Apresentados

Mídia: CD-ROM  
 Manuais: MANUAL DO USUÁRIO  
 Outros: TECLADO SEM-FIO

Características do Equipamento

Hardware: DURON 1,2GB Monitor: AOC 15"  
 Hardware Especial: SCANNER Software Especial: INTERNET EXPLORER  
 Espaço em Disco: 16GB Memória RAM: 256 MB  
 Impressora: HP 656C Sist. Operacional: WINDOWS XP

Data Inicial da Avaliação: 10/7/2003  
 Data Final da Avaliação: 10/7/2003

Observações  
 AVALIAÇÃO P/ AQUISIÇÃO DE UM SOFTWARE FINANCEIRO DE ACORDO COM A NORMA ISO/IEC 9126

Questões	Peso	Opção
01.01 - Adequação O software possui todas as funções no produto?	25,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input checked="" type="checkbox"/> Completamente
01.01 - Adequação O software dispõe de todas as funções necessárias para a sua execução?	25,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input checked="" type="checkbox"/> Completamente

0% Page 1 of 2

Impressão da Avaliação

02.01 - Maturidade O software tem capacidade de garantir os dados na queda de energia durante o processamento de atualização de dados?	15,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente
---	-------	--

Resultado das Característica	Peso	Resultado	%
01.00 Funcionalidade	60,00	47,7273	79,55 %
01.01 Adequação	25,00	25,0000	100,00 %
01.02 Acurácia	30,00	18,7500	62,50 %
02.00 Confiabilidade	40,00	30,0000	75,00 %
02.01 Maturidade	15,00	11,2500	75,00 %

Legenda da Classificação

90.60% à 100.00% - Excelente (Aceito)  
 75.60% à 90.59% - Bom (Aceito)  
 50.60% à 75.59% - Regular (Necessita de Revisão)  
 00.00% à 50.59% - Insuficiente (Rejeitado)

Classificação Geral: **Bom - Aceito** Resultado Geral: **77.73 %**

0% Page 2 of 2

# APÊNDICE C – RELATÓRIO DE COMPARAÇÃO DE AVALIAÇÃO DE DOIS PRODUTOS DE SOFTWARE

Comparativo de Avaliações				
<b>Avaliação 1: Finanças 1 - NBR ISO/IEC 12119</b>				
<b>Avaliação 2: Finanças 3 - NBR ISO/IEC 12119</b>				
Questões	Peso 1	Opção 1	Peso 2	Opção 2
07.01 - Req. sobre o Conteúdo da Descr O Produto possui uma Descrição do Produto?	60,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	60,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input checked="" type="checkbox"/> Completamente
07.01 - Req. sobre o Conteúdo da Descr A Descrição do Produto é suficientemente compreensiva?	60,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	60,00	<input type="checkbox"/> Não Atende <input checked="" type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input type="checkbox"/> Completamente
07.01 - Req. sobre o Conteúdo da Descr É fácil ter uma visão geral pela Descrição do Produto?	60,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	60,00	<input checked="" type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input type="checkbox"/> Completamente
07.01 - Req. sobre o Conteúdo da Descr A Descrição de Produto está livre de inconsistências internas?	60,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	60,00	<input type="checkbox"/> Não Atende <input checked="" type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input type="checkbox"/> Completamente
07.01 - Req. sobre o Conteúdo da Descr O significado dos termos utilizados permanece constante em toda a Descrição de Produto?	60,00	<input type="checkbox"/> Não Atende <input checked="" type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input type="checkbox"/> Completamente	60,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente
07.02 - Identificações e Indicações O Produto possui uma identificação da Descrição do Produto?	30,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	30,00	<input type="checkbox"/> Não Atende <input checked="" type="checkbox"/> Parcialmente <input type="checkbox"/> Largamente <input type="checkbox"/> Completamente
09.05 - Eficiência Caso existam declarações sobre eficiência na Descrição do Produto, elas estão em conformidade com o produto?	40,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	40,00	<input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente <input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente
09.06 - Manutenibilidade Caso existam declarações sobre manutenibilidade na Descrição do Produto, elas estão em conformidade com o produto?	30,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	30,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente
09.07 - Portabilidade Caso existam declarações sobre portabilidade na Descrição do Produto, elas estão em conformidade com o produto?	30,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente	30,00	<input type="checkbox"/> Não Atende <input type="checkbox"/> Parcialmente <input checked="" type="checkbox"/> Largamente <input type="checkbox"/> Completamente
<b>Legenda da Classificação</b>				
90.60% à 100.00% - Excelente (Aceito)				
75.60% à 90.59% - Boa (Aceito)				
50.60% à 75.59% - Regular (Necessita de Revisão)				
00.00% à 50.59% - Insuficiente (Rejeitado)				
<b>Resultado Geral:</b>		<b>68,82 %</b>		<b>75,39 %</b>

TCC - Fabiano Stimariglio Ferreira

## APÊNDICE D – RELATÓRIO DE AVALIAÇÃO EM BRANCO

Impressão da Avaliação em Branco

Identificação da Avaliação: Avaliação Financeira

Avaliador: \_\_\_\_\_  
 Produtor: \_\_\_\_\_  
 Produto: \_\_\_\_\_ Versão: \_\_\_\_\_  
 Informações do Produto

Materiais Apresentados

Mídia: \_\_\_\_\_  
 Manuais: \_\_\_\_\_  
 Outros: \_\_\_\_\_

Características do Equipamento

Hardware: \_\_\_\_\_ Monitor: \_\_\_\_\_  
 Hardware Especial: \_\_\_\_\_ Software Especial: \_\_\_\_\_  
 Espaço em Disco: \_\_\_\_\_ Memória RAM: \_\_\_\_\_  
 Impressora: \_\_\_\_\_ Sist. Operacional: \_\_\_\_\_

Data Inicial da Avaliação: \_\_\_/\_\_\_/\_\_\_  
 Data Final da Avaliação: \_\_\_/\_\_\_/\_\_\_

Observações

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

**Questões**

<b>01.01 - Adequação</b>	[ ] Não Atende
O software possui todas as funções no produto?	[ ] Parcialmente
	[ ] Largamente
Peso: _____	[ ] Completamente
<b>01.01 - Adequação</b>	[ ] Não Atende
O software dispõe de todas as funções necessárias para a sua execução?	[ ] Parcialmente
	[ ] Largamente
Peso: _____	[ ] Completamente
<b>01.01 - Adequação</b>	[ ] Não Atende
O software dispõe de funções para processamento de rotinas específicas?	[ ] Parcialmente
	[ ] Largamente
Peso: _____	[ ] Completamente

0% Page 1 of 2