

Universidade de Brasília - UnB
Faculdade UnB Gama - FGA
Engenharia de Software

Uma proposta de método de escolha para sistemas CMSs

Autor: Thiago Silveira Honorato
Orientador: Prof. Msc. Ricardo Ajax Dias Koloski

Brasília, DF
2015



Thiago Silveira Honorato

Uma proposta de método de escolha para sistemas CMSs

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Msc. Ricardo Ajax Dias Koloski

Brasília, DF

2015

Thiago Silveira Honorato

Uma proposta de método de escolha para sistemas CMSs/ Thiago Silveira Honorato. – Brasília, DF, 2015-

166 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Msc. Ricardo Ajax Dias Koloski

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2015.

1. Sistemas Gerenciadores de Conteúdo. 2. Método. I. Prof. Msc. Ricardo Ajax Dias Koloski. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Uma proposta de método de escolha para sistemas CMSs

CDU 02:141:005.6

Thiago Silveira Honorato

Uma proposta de método de escolha para sistemas CMSs

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 02 / 07 /2015:

Prof. Msc. Ricardo Ajax Dias Koloski
Orientador

Profa. Msc. Fabiana Freitas Mendes
Convidado 1

**Prof. Dr. Wander Cleber Maria
Pereira da Silva**
Convidado 2

Brasília, DF
2015

Agradecimentos

Agradeço a Deus pelo dom da vida, pela saúde e por ter chegado até aqui. Agradeço aos meus pais Evando e Marly e a toda minha família pelo apoio dado ao longo de todos esses anos de caminhada escolar e acadêmica. Agradeço a todos os meus professores que durante anos me incentivaram a acreditar sempre em mim e ir em busca dos meus sonhos. Agradeço ao meu orientador professor Ricardo Ajax pela paciência e contribuição ao trabalho com boas ideias. Agradeço aos meus colegas de curso, aos meus amigos e a minha namorada por toda força dada ao longo desses 5 anos de graduação.

Resumo

Qualquer conteúdo para ser divulgado e compartilhado na web, requer atenção especial das linguagens de programação a serem utilizadas, desde as mais simples até as mais complexas, dependendo do conteúdo a ser disponibilizado. O conhecimento necessário costuma ter uma determinada complexidade que pode não ser comum a usuários leigos. Esta situação pode ser amenizada pelo uso de sistemas de gestão de conteúdo (CMS – Content Management System) que, por sua vez, são plataformas que fazem a união entre diversos mecanismos que permitem criar e publicar conteúdo em tempo real. Porém a escolha de um determinado CMS pode ser algo não muito fácil, pois tais sistemas permitem serem utilizados em diversos contextos e situações. Assim sendo, a escolha de um CMS pode ser facilitada com um método claro a ser seguido por aqueles que necessitarem tomar a decisão de qual CMS usar no seu contexto específico. O método proposto neste trabalho leva em consideração as principais características encontradas em sistemas do tipo CMS, organizando um procedimento que conduza o usuário a uma escolha que lhe seja vantajosa, pois esses sistemas podem fornecer o apoio necessário para a construção de redes sociais, sites de e-Commerce, revistas eletrônicas, entre outras possibilidades.

Palavras-chaves: CMS. Método. Desenvolvimento Web. Qualidade de Software. SQuaRE. Características.

Abstract

Any content to be made available and shared on the web, requires special attention from programming languages to be used, from the simplest to the most complex, depending on the content to be made available. The knowledge required usually has a certain complexity that may not be common to lay users. This situation can be mitigated by the use of systems of content management (CMS - Content Management System), which, in turn are platforms that make the union between various mechanisms that allow you to create and publish content in real time. But the choice of a particular CMS can be something not so easy, why such systems allow to be used in a variety of contexts and situations. Thus, the choice of a CMS can be facilitated with a clear method to be followed by those who will take the decision of which CMS have to be use in their specific contexts. The method proposed in this work takes into account the main features found in type systems CMS, organizing a procedure that will lead the user to a choice that is advantageous, because these systems can provide the necessary support for the construction of: social networks, e-Commerce sites, electronic journals, among other possibilities.

Keywords: CMS. Method. Web Development. Software Quality. SQuaRE. Features.

Lista de ilustrações

Figura 1 – Estrutura Analítica do Projeto de TCC.	29
Figura 2 – Classificação de Requisitos Não Funcionais de (MAMANI, 1999)	33
Figura 3 – Classificação de Requisitos não Funcionais por SOMMERVILLE (1992). 33	
Figura 4 – Popularidade pelo número de commits no git hub.	38
Figura 5 – Percentual de popularidade pelo número de commits no git hub.	38
Figura 6 – Popularidade de acordo com a rede social Facebook.	39
Figura 7 – Percentual de popularidade de acordo com a rede social Facebook. . . .	39
Figura 8 – Popularidade de acordo com a rede social Twitter.	39
Figura 9 – Percentual de popularidade de acordo com a rede social Twitter.	40
Figura 10 – Ranking de melhor desempenho no experimento 1	44
Figura 11 – Ranking de melhor desempenho no experimento 2	44
Figura 12 – Ranking de melhor desempenho no experimento 3	44
Figura 13 – Organização da série SQuaRE (ISO/IEC-25000, 2005)	47
Figura 14 – Estrutura do modelo de qualidade da SQuaRE (ISO/IEC-25010, 2011) 49	
Figura 15 – Qualidade em Uso segundo a SQuaRE (ISO/IEC-25010, 2011)	50
Figura 16 – Qualidade do Produto segundo a SQuaRE (ISO/IEC-25010, 2011) . . .	52
Figura 17 – Fases do GQM (SOLLINGEN; BERGHOUT, 1999)	58
Figura 18 – Níveis do GQM (SOLLINGEN; BERGHOUT, 1999)	59
Figura 19 – Estrutura do GQM (BASILI; CALDIERA; ROMBACH, 1994)	60
Figura 20 – Construtor de Métricas (MCGARRY et al., 2002),(BORGES, 2003) . .	63
Figura 21 – Modelo de Informação do PSM (CARD; JONES, 2003)	64
Figura 22 – Resultado da Questão 1 para o Questionário 1	70
Figura 23 – Resultado da Questão 2 para o Questionário 1	70
Figura 24 – Resultado da Questão 3 para o Questionário 1	71
Figura 25 – Resultado da Questão 3 por percentual para o Questionário 1.	71
Figura 26 – Resultado da Questão 4 para o Questionário 1	72
Figura 27 – Resultado da Questão 5 para o Questionário 1	72
Figura 28 – Resultado da Questão 6 para o Questionário 1	73
Figura 29 – Resultado da Questão 7 para o Questionário 1	73
Figura 30 – Resultado da Questão 8 para o Questionário 1	74
Figura 31 – Resultado da Questão 9 para o Questionário 1	74
Figura 32 – Resultado da Questão 10 para o Questionário 1	75
Figura 33 – Resultado da Questão 11 para o Questionário 1	76
Figura 34 – Resultado da Questão 12 para o Questionário 1	76
Figura 35 – Resultado da Questão 13 para o Questionário 1	77
Figura 36 – Fluxograma do método proposto	79

Figura 37 – Diagrama GQM para a Qualidade do Produto	89
Figura 38 – Diagrama GQM para Qualidade em Uso	89
Figura 39 – Gráfico de declividade para a matriz de dados	92
Figura 40 – Frequência de respostas para a métrica de operacionalidade	95
Figura 41 – Frequência de respostas para a métrica de estética	95
Figura 42 – Frequência de respostas para a métrica de adaptabilidade	96
Figura 43 – Frequência de respostas para a métrica de modificabilidade	96
Figura 44 – Frequência de respostas para a métrica de reusabilidade	96
Figura 45 – Frequência de respostas para a métrica de modularidade	97
Figura 46 – Frequência de respostas para a métrica de testabilidade	97
Figura 47 – Frequência de respostas para a métrica de comportamento do tempo	97
Figura 48 – Frequência de respostas para a métrica de Utilização de Recursos	97
Figura 49 – Frequência de respostas para a métrica de maturidade	98
Figura 50 – Frequência de respostas para a métrica de tolerância a falhas	98
Figura 51 – Frequência de respostas para a métrica de recuperabilidade	98
Figura 52 – Frequência de respostas para a métrica de efetividade	98
Figura 53 – Frequência de respostas para a métrica de flexibilidade	99
Figura 54 – Rastreabilidade de Características para CMSs populares Parte 1	136
Figura 55 – Rastreabilidade de Características para CMSs populares Parte 2	137
Figura 56 – Rastreabilidade de Características para CMSs não populares Parte 1	138
Figura 57 – Rastreabilidade de Características para CMSs não populares Parte 2	138

Lista de tabelas

Tabela 1 – Mapeamento de CMSs em Software Livre e Software Proprietário	36
Tabela 2 – Popularidade dos CMSs Mapeados	38
Tabela 3 – Vantagens e Desvantagens dos CMSs, segundo (COSTA; CAMINHA; LIMA, 2012), (REIS et al., 2011)	42
Tabela 4 – Características avaliadas pelo experimento de Patel, Rathod e Parikh (2011)	43
Tabela 5 – Ranking gerado a partir do experimento de Patel, Rathod e Parikh (2011).	44
Tabela 6 – SQuaRE X ISO 9126 (ISO/IEC-25010, 2011)	53
Tabela 7 – Tipos de métricas de Software.	56
Tabela 8 – Tipos de escalas de métricas de Software Fonte: (PARK; GOETHERT; FLORAC, 1996, p. 9); (FENTON; PFLEGEER, 1997, pp. 45-53)	57
Tabela 9 – Detalhamento das necessidades da informação do PSM.(BAILEY et al., 2003)	61
Tabela 10 – Conceitos Mensuráveis do PSM.	62
Tabela 11 – Cruzamento de Características aprovadas pelos especialistas com ca- racterísticas SquaRE - Qualidade do Produto.	83
Tabela 12 – Cruzamento de Características aprovadas pelos especialistas com ca- racterísticas SquaRE - Qualidade em uso.	85
Tabela 13 – Tomada de decisão	87
Tabela 14 – Objetivo de Medição (SOLLINGEN; BERGHOUT, 1999).	87
Tabela 15 – Objetivo de Medição (SOLLINGEN; BERGHOUT, 1999).	88
Tabela 16 – Resultados para os indicadores estabelecidos	92
Tabela 17 – Cargas fatoriais para um fator	93
Tabela 18 – Média e Desvio Padrão para questões com carga fatorial insuficiente. . .	94
Tabela 19 – Outras medidas para as métricas mais relevantes	99
Tabela 20 – Outras medidas para as métricas mais relevantes	99
Tabela 21 – Contexto para artigos levantados.	119
Tabela 22 – Contexto para artigos levantados CMSs não populares.	123
Tabela 23 – Características de CMSs levantadas para CMSs populares.	125
Tabela 24 – Características de CMSs levantadas para CMSs não populares.	130
Tabela 25 – Cruzamento de características de CMS com características SQuaRE - Qualidade do Produto.	132
Tabela 26 – Cruzamento de características de CMS com características SQuaRE - Qualidade em Uso	135
Tabela 27 – M1 - Cobertura Funcional - Adaptado de ISO/IEC-25023 (2011)	148

Tabela 28 – M2 - Percentual de mensagens claras - Adaptado de ISO/IEC-25023 (2011)	148
Tabela 29 – M3 - Documentação Plena - Adaptado de ISO/IEC-25023 (2011)	149
Tabela 30 – M4 - Percentual de elementos personalizáveis - Adaptado de ISO/IEC-25023 (2011)	150
Tabela 31 – M5 - Controle de acesso - Adaptado de ISO/IEC-25023 (2011)	150
Tabela 32 – M6 - Prevenção a dados corrompidos - Adaptado de ISO/IEC-25023 (2011)	151
Tabela 33 – M7 - Mudança de configuração - Adaptado de (ISO/IEC-9126-2, 2001)	151
Tabela 34 – M8 - Complexidade de modificação - Adaptado de ISO/IEC-25023 (2011)	152
Tabela 35 – M9 - Condensabilidade - Adaptado de ISO/IEC-25023 (2011)	153
Tabela 36 – M10 - Percentual de módulos reusáveis - Adaptado de ISO/IEC-25023 (2011)	153
Tabela 37 – M11 - Completude de testes de sistema - Adaptado de ISO/IEC-25023 (2011)	154
Tabela 38 – M12 - Vazão - Adaptado de ISO/IEC-25023 (2011)	154
Tabela 39 – M13 - Utilização de memória - Adaptado de ISO/IEC-25023 (2011)	155
Tabela 40 – M14 - Remoção de Falhas - Adaptado de ISO/IEC-25023 (2011)	155
Tabela 41 – M15 - Tempo de Serviço - Adaptado de ISO/IEC-25023 (2011)	156
Tabela 42 – M16 - Percentual de módulos redundantes instalados - Adaptado de ISO/IEC-25023 (2011)	156
Tabela 43 – M17 - Recuperação após falha - Adaptado de ISO/IEC-25023 (2011)	157
Tabela 44 – M18 - Percentual de tarefas completas - Adaptado de ISO/IEC-25022 (2012)	157
Tabela 45 – M19 - Características flexíveis - Adaptado de ISO/IEC-25022 (2012)	158
Tabela 46 – Questionário checklist para a execução da métrica de Completude Funcional.	159
Tabela 47 – Parâmetros de corretude gramatical	160
Tabela 48 – Textos curtos	161
Tabela 49 – Terminologia	161
Tabela 50 – Cargas fatoriais para seis fatores	162
Tabela 51 – Cargas fatoriais para cinco fatores	163
Tabela 52 – Cargas fatoriais para quatro fatores	163
Tabela 53 – Cargas fatoriais para três fatores	164
Tabela 54 – Cargas fatoriais para dois fatores	165

Sumário

1	INTRODUÇÃO	19
1.1	Motivação e justificativa	19
1.1.1	Plataformas de Grande Porte	19
1.1.2	Desenvolvimento Cliente Servidor	20
1.1.3	Desenvolvimento Web	20
1.2	Tema	21
2	DEFINIÇÕES	23
2.1	Problema Abordado Dentro do Tema	23
2.1.1	Questão de Pesquisa	23
2.2	Objetivos	24
2.2.1	Objetivo Geral	24
2.2.2	Objetivos Específicos	24
2.3	Metodologia de Pesquisa	24
2.4	Revisão de Literatura	25
2.4.1	Bases de dados consultadas	26
2.4.2	Expressões de Busca elaboradas	26
2.4.3	Critérios de Seleção	26
2.5	Coleta de Dados	26
2.5.1	População, amostra e sujeitos	26
2.5.2	Instrumentos de Coleta de Dados	27
2.5.3	Procedimentos de Coletas de dados	28
2.6	Planejamento	28
2.7	Organização da Pesquisa	30
3	CMS	31
3.1	Revisão de literatura para o capítulo	31
3.2	Definições Básicas	31
3.2.1	CMSs e suas características	32
3.2.1.1	Características gerais (Requisitos) de CMSs	32
3.2.2	Vantagens no uso de CMS para o desenvolvimento web	33
3.2.3	Desvantagens no uso de CMS para o desenvolvimento web	34
3.3	Uso de CMS na indústria de Software	34
3.4	Critérios para a escolha de CMSs e escolha de CMSs candidatos	35
3.4.1	Software Livre / Software Proprietário	35
3.4.2	Popularidade	37

3.5	Vantagens e Desvantagens dos CMS mais populares	41
3.6	Comparativos já Realizados	43
4	QUALIDADE DE SOFTWARE	46
4.1	Revisão de literatura para o capítulo	46
4.2	As Normas SQuaRE	46
4.3	Organização das normas Square	47
4.4	Qualidade de Software segundo a SQuaRE	48
4.4.1	O Modelo de Qualidade em Uso da Square	49
4.4.2	O Modelo de Qualidade do Produto segundo a norma Square	49
4.4.3	Comparação entre a SQuaRE e a ISO / IEC 9126	53
5	MÉTRICAS DE SOFTWARE	55
5.1	Revisão de literatura para o capítulo	55
5.2	Tipos de Métricas de Software	55
5.3	GQM	58
5.4	PSM	60
6	APLICAÇÃO	66
6.1	Identificação das características	66
6.1.1	Características de CMSs para os mais populares	66
6.1.2	Características de CMSs para os não populares	67
6.1.3	Rastreabilidade de características	67
6.1.4	Contagem de características e mapeamento com as características SQuaRE	68
6.2	Aplicação do Primeiro Questionário	69
6.2.1	Critérios para análise de resultados do primeiro questionário	69
6.2.2	Análise dos resultados das questões do questionário	70
6.3	Visão Geral do Método	78
6.4	Construção do Método - Parte I	80
6.4.1	Listagem de características aprovadas pelos especialistas	80
6.4.2	Passo a passo para a escolha do CMS - Parte I	80
6.5	Construção do Método - Parte II	82
6.5.1	Listagem de características aprovadas pelos especialistas	83
6.5.2	Passo a passo para a escolha do CMS - Parte II	85
6.5.3	Objetivo de Medição	87
6.5.4	Questões	88
6.5.5	Métricas	89
6.6	Análise de resultados para o segundo questionário	90
6.6.1	Procedimentos necessários para a execução da Análise de Fatores	90
6.6.2	Resultados obtidos - Validação da matriz de dados	91

6.6.3	Extração de fatores	92
6.6.4	Breve estatística descritiva para as Métricas mais significantes	95
7	CONCLUSÕES	101
7.1	Sugestões de Trabalhos Futuros	103
	Referências	105
	 APÊNDICES	 115
	APÊNDICE A – DETALHAMENTO DAS EXPRESSÕES DE BUSCA	
	ELABORADAS	116
A.1	Resultados para os CMS mais populares	117
A.2	Resultados para os CMS não populares	117
A.3	Critérios de Exclusão para os artigos selecionados	118
	 APÊNDICE B – LEVANTAMENTO DE CARACTERÍSTICAS DE	
	CMS - TABELAS	119
B.1	Contexto de artigos Lidos	119
B.1.1	Contexto para artigos envolvendo CMSs mais populares	119
B.1.2	Contexto para artigos envolvendo CMSs não populares	123
B.2	Características de CMSs mais populares	125
B.3	Características de CMSs não populares	130
B.4	Cruzamento de Características de CMS com características SQuaRE	132
B.5	Rastreabilidade de Características - Figuras	135
	 APÊNDICE C – QUESTIONÁRIOS	 139
C.1	Questionário 1 - Levantamento de Características de CMSs	139
C.2	Questionário 2 - Validação do Método Proposto	143
	 APÊNDICE D – GQM - MÉTRICAS DEFINIDAS	 148
D.1	Métricas - Adequação Funcional	148
D.2	Métricas - Usabilidade	148
D.3	Métricas - Segurança	150
D.4	Métricas - Portabilidade	151
D.5	Métricas - Manutenibilidade	152
D.6	Métricas - Eficiência de Desempenho	154
D.7	Métricas - Confiabilidade	155
D.8	Métricas - Efetividade	157
D.9	Métricas - Cobertura de Contexto	158

	APÊNDICE E – SUPORTE PARA AS MÉTRICAS DEFINIDAS . .	159
E.1	Compleitude Funcional	159
E.2	Operacionalidade	160
	APÊNDICE F – CARGAS FATORIAIS	162

1 Introdução

Desde os primórdios da década de 50 e 60, a capacidade de armazenar, processar dados e informações de forma eletrônica evoluiu bastante. Se, por um lado, inicialmente o armazenamento e processamento de dados era restrito a um público técnico altamente especializado, atualmente existe uma série de ferramentas de apoio ao desenvolvimento de software voltados para o ambiente composto pela Internet, que servem aos mais variados propósitos, desde a simples divulgação de informações pessoais estáticas, perpassando as redes sociais com seus vários mecanismos de troca de informações, até chegar ao comércio eletrônico, no qual vários tipos de apoio a negócios são feitos nas mais diversas áreas da atividade humana.

Este capítulo procura motivar e justificar o estudo deste trabalho de conclusão de curso contemplando o contexto em que se inserem os CMSs.

1.1 Motivação e justificativa

Nesta seção são apresentadas definições básicas de plataformas de grande porte, sistema cliente-servidor e desenvolvimento web com o intuito de motivar e justificar a escolha do tema a ser apresentado na Seção 1.2.

1.1.1 Plataformas de Grande Porte

Um exemplo conhecido de desenvolvimento em grande porte é o uso de Mainframes, conhecidos por sua confiabilidade, estabilidade, segurança, disponibilidade e grande capacidade de processamento de dados. Os componentes de hardware de um mainframe têm capacidade de autocontrole e auto recuperação vastas. A confiabilidade do software do sistema é resultado de imensos testes e da capacidade de fazer atualizações rápidas para os problemas detectados (SHU et al., 2013).

Além do uso de mainframes, é importante destacar também os sistemas monolíticos, que, para Tanenbaum e Woodhull (2006, p. 58), é uma “grande bagunça”, pois consiste apenas em um conjunto de rotinas, no qual cada rotina pode chamar qualquer uma das outras conforme a necessidade. Apesar da enorme capacidade de processamento, sistemas monolíticos são bastantes sensíveis a erros, pois, devido ao fato das rotinas poderem interagir com outras rotinas de forma muito fácil, quando há um erro, este erro se alastra rapidamente por todo sistema e assim pode causar a interrupção desse sistema.

Uma vantagem dos sistemas monolíticos é que a atualização ocorre de forma simples, pois basta atualizar a máquina central e, como os terminais são sem capacidade de

processamento local, toda a rede estará atualizada (TANENBAUM; WOODHULL, 2006, p. 58).

1.1.2 Desenvolvimento Cliente Servidor

De acordo com Tanenbaum e Woodhull (2006, p. 58), os sistemas monolíticos eram desprovidos de grandes formalismos em suas estruturas, podendo pedaços de código (funções) chamar outros pedaços (funções) sem grandes restrições.

Com uma das motivações de reduzir a enorme quantidade de funções executadas pelo sistema operacional, foi proposto o sistema cliente servidor que, para Tanenbaum e Woodhull (2006, p. 64), implementa a maior parte dos procedimentos do sistema operacional na forma de processos de usuário. Tanenbaum e Woodhull (2006, p. 64), ainda acrescentam que no momento de solicitar um serviço, como executar uma determinada instrução de leitura, o processo de usuário (agora chamado de processo cliente) envia uma requisição para um processo servidor que então executa a tarefa e fornece uma resposta.

Para Duchessi e Chengalur-Smith (1998), o sistema Cliente - Servidor apresenta inúmeros benefícios. Alguns desses benefícios são:

- Melhoria da acessibilidade do sistema.
- Redução de custos.
- Aumento da produtividade organizacional.
- Aumento na confiabilidade do Sistema.

Apesar de evitar a interrupção do sistema diante de um erro em alguma rotina, o sistema cliente - servidor apresenta alguns problemas. Um desses problemas é o fato de que, como existe processamento por parte do cliente, cada vez que se atualiza um sistema, todos os clientes devem ser atualizados, mesmo que isso seja feito de forma automática, o que foi uma das motivações para o surgimento dos sistemas web.

1.1.3 Desenvolvimento Web

De acordo com Junior e Antônio (2003), a tecnologia Web funciona de forma relativamente simples, pois a Web é um conjunto de arquivos digitais que ficam armazenados em servidores ligados a Internet e que podem ser recuperados a partir de qualquer computador que tenha conexão com a Internet. Estes arquivos digitais são as chamadas páginas web que compõem a grande rede de informações que é a World Wide Web.

Os mesmos autores ainda afirmam que, com o passar do tempo, a Web foi adquirindo novos recursos e novas funções e houve um grande progresso quando os seus usuários

passaram a não só solicitar páginas com conteúdo estático, mas também retornar resultados com conteúdo dinâmico.

Hochberger e Meusel (2008) dizem que uma dimensão importante para páginas web é manter um modelo consistente ao longo de todas as páginas. Aplicações web comuns consistem em elementos como um cabeçalho, um menu e um conteúdo da página agrupados em um esqueleto de página. Possuir uma separação estrutural em pedaços de HTML comuns e de código permite que tais pedaços possam ser compartilhados entre todas as páginas, o que traz um reuso para uma aplicação web de forma significativa.

Os sistemas web apresentam várias vantagens em relação aos demais sistemas Connallen (2003 apud PERIZZOLO, 2005), diz que uma dessas vantagens é a manutenção, desenvolvimento e atualização centrada na aplicação. Isto é, diferente do sistema cliente servidor em que todos os clientes têm que ser atualizados quando o servidor é atualizado, nos sistemas web basta o servidor ser atualizado para que os clientes obtenham acesso as atualizações, reduzindo, assim, o custo de implementação das operações. Entretanto os sistemas web apresentam desvantagens. Uma desvantagem é que não há padronização para os navegadores de internet, o que pode acarretar em problemas de incompatibilidade dependendo de qual navegador está sendo usado e de qual versão é tal navegador (CONNALLEN, 2003 apud PERIZZOLO, 2005, p. 144).

1.2 Tema

Coelho et al. (2011) afirmam que qualquer conteúdo para ser divulgado e compartilhado na web, requer atenção especial das linguagens de programação a serem utilizadas. Na maioria das vezes o nível de complexidade para a construção de páginas web é alto e inacessível para leigos. Com esta motivação, em 1990 surgiram os sistemas de gestão de conteúdo (*Content Management System*- CMS) que, para BARCIA (2008), são plataformas que fazem a união entre diversos mecanismos os quais permitem criar e publicar conteúdo em tempo real em que, os usuários não precisam dominar uma programação específica para a construção de páginas na web, e além disso, possuem contato com páginas intuitivas e dinâmicas.

Barrére, Avelino e Eduardo (S/D), declaram que sistemas CMSs oferecem diversos outros recursos. Alguns exemplos desses recursos são a fácil publicação e navegação por meio do conteúdo divulgado e a classificação de conteúdos em categorias. Além disso, pode facilmente agregar funcionalidades como blogs, ou fóruns de discussão.

Por oferecerem uma grande variedade de recursos, os CMSs são uma boa opção para quem deseja construir uma página web, pois permite que o desenvolvedor já inicie o projeto da sua página a partir de um determinado template. Nath e Arora (2010) ilustram que o processo de escolha de um CMS pode não ser algo simples, pois, além de

existirem vários CMSs no mercado, cada CMS possui características únicas que podem ser decisivas para a montagem de uma aplicação específica, isto é, a escolha de um CMS pode influenciar em aspectos funcionais, de segurança, de desempenho, entre outros. Por esse motivo a criação de um método que auxilie o desenvolvedor a escolher um CMS faz-se necessário.

2 Definições de Estudo

Neste capítulo são definidos o problema a ser trabalhado, os objetivos gerais e específicos, a classificação da pesquisa, e o planejamento de como o trabalho será executado.

2.1 Problema Abordado Dentro do Tema

Para Filho (2003), o valor de um produto vem de suas características que, se tratando de software, podem ser funcionais ou não funcionais. Características funcionais representam os comportamentos que o sistema deve apresentar diante das ações de seus usuários. Características não funcionais definem determinados aspectos do comportamento que um sistema deve ter ao executar suas características funcionais. Filho (2003) acrescenta que os requisitos funcionais explicam as funcionalidades que o produto deverá realizar em benefício daqueles que usam o sistema, enquanto que a especificação dos requisitos não funcionais completa os requisitos funcionais, descrevendo as restrições e outras características que o sistema deve possuir para que seja atingida a qualidade desejada.

Uma forma de enunciar objetivamente grandezas é a partir de medições. Neste sentido, a norma ISO 25000 expressa características em uso do produto, ou do próprio produto por meio de medições (ISO/IEC-25000, 2005).

Uma forma de construir sistemas web é com o uso de Sistemas Gerenciadores de Conteúdo ou *Content Management System* (CMS), que permitem ao usuário construir sites para a internet com auxílio de vários modelos pré-concebidos de páginas web e diversos componentes pré-construídos. Seu uso concede ao usuário que deseja desenvolver páginas para internet flexibilidade, pois é possível que esse usuário separe o conteúdo da página do seu design, dentre outras vantagens. Como alguns exemplos disponíveis na indústria de software podem ser citados: Wordpress, Drupal e Joomla (COSTA; CAMINHA; LIMA, 2012).

A partir da motivação e tema propostos a questão de pesquisa é apresentada na Seção 2.1.1, que vem logo a seguir.

2.1.1 Questão de Pesquisa

A questão de pesquisa abordada por este trabalho é:

Como estabelecer um método objetivo para a escolha de CMSs a serem utilizados no desenvolvimento de sites Web?

2.2 Objetivos

Esta seção apresenta os objetivos geral e específicos do trabalho a ser desenvolvido.

2.2.1 Objetivo Geral

Propor um método objetivo que possa servir de apoio para a escolha de um CMS.

2.2.2 Objetivos Específicos

Para o Trabalho de Conclusão de Curso (TCC) pretende-se cumprir com os seguintes objetivos específicos.

1. Identificar o que é um CMS.
2. Examinar conceitos de qualidade de software, a fim de esclarecer como medir as características de sistemas CMSs, a fim de permitir que elas sejam mensuradas.
3. Definir medições para avaliar as características de CMSs, a partir de conceitos das teorias para estabelecer medições usadas na Engenharia de Software.
4. Identificar sistemas de CMSs candidatos para serem usados como estudos de caso na aplicação do método proposto.
5. Estabelecer um método que possa ser aplicado aos CMSs candidatos, a partir de medições oriundas de conceitos de qualidade de software.
6. Validar o método proposto a partir de opiniões de especialistas no assunto.

2.3 Metodologia de Pesquisa

Segundo [Moresi \(2003\)](#), a pesquisa a ser realizada para o Trabalho de Conclusão de Curso(TCC) possui como principais características:

Quanto à natureza: é uma *Pesquisa Aplicada*, pois aplica conceitos para que o conhecimento gerado venha a ser utilizado para a realização de práticas que possam ser dirigidas à solução de problemas específicos.

Quanto à forma de abordagem do problema: é uma pesquisa *Mista*, pois utiliza-se de técnicas qualitativas e quantitativas para a sua construção. Foi construído um método baseado em métricas de software para ajudar na seleção de produtos do tipo CMS. Este método foi validado com o auxílio de especialistas e esta validação foi apoiada por métodos estatísticos.

Quanto aos fins: a pesquisa é *Metodológica*, pois seu objetivo é construir um método apoiado por métricas de software para facilitar a escolha de um determinado CMS. Além disso, a pesquisa deve ser *Bibliográfica*, pois o estudo foi desenvolvido com base em livros, outras monografias, dissertações, teses, artigos, além do próprio uso da internet como apoio para a realização da pesquisa.

Quanto a forma de coleta de dados: Foram realizados *surveys* com especialistas e usuários para refinar o conjunto de características que foram medidas e validar o método proposto.

2.4 Revisão de Literatura

Tendo em vista a questão de pesquisa postulada na Seção 2.1.1: "Como estabelecer um método objetivo para a escolha de CMSs a serem utilizados no desenvolvimento de sites Web?", os objetivos geral e específicos apresentados nas seções 2.2.1 e 2.2.2 respectivamente, foi feita uma revisão de literatura.

Inicialmente foi importante identificar o que são sistemas CMS, quais as vantagens e desvantagens de seu uso na indústria de software, além de apresentar um estudo prévio, com o objetivo de mostrar quais os CMSs mais populares. Em seguida foi importante identificar quais seriam as características relevantes para serem abordadas pelo método proposto no objetivo geral deste trabalho.

Dando continuidade à revisão de literatura deste TCC, conforme os objetivos já citados foram avaliados conceitos de qualidade baseados na série de normas ISO 25000. Além disso, foi necessário um estudo a respeito da teoria de medições comumente usada na engenharia de software. Esse estudo foi realizado para, a partir das medições definidas, tornar o método o mais objetivo possível.

Para fortalecer a revisão de literatura foram usados princípios da técnica de revisão sistemática de literatura, compreendendo os seguintes itens.

- Uso de **bases digitais** de publicações.
- Uso de **expressões de busca** para serem aplicadas às bases digitais de forma a automatizar a busca e recuperação de publicações.
- Uso de **critérios de seleção** para a escolha de publicações relevantes para o estudo, a partir dos resultados obtidos com as expressões de busca.
- Utilização de princípios de **coletas de dados**.

2.4.1 Bases de dados consultadas

A partir de uma revisão prévia de literatura, foram identificadas as bases de dados nas quais seriam pesquisadas publicações sobre o assunto:

1. IEEE.
2. ACM.
3. Scopus.
4. Science Direct.
5. Springer.
6. Google Scholar.

2.4.2 Expressões de Busca elaboradas

Para o levantamento de características apresentado na Seção 6.1 foram elaboradas expressões de busca (*strings de pesquisa*) a fim de obter uma maior profundidade em torno das características de CMS. A expressão de busca foi construída a partir de palavras chave comumente encontradas nas publicações obtidas na revisão prévia de literatura. O detalhamento das expressões de busca e dos resultados encontrados está no Apêndice A deste TCC.

2.4.3 Critérios de Seleção

Foram estabelecidos critérios de inclusão/exclusão de artigos como forma de filtrar as publicações mais relevantes sobre o tema, problema e objetivos deste trabalho de TCC.

Os critérios de seleção gerais são apresentados na primeira seção de cada capítulo. Porém os critérios utilizados para a exclusão de artigos usados para a seleção de características são apresentados no apêndice A.

2.5 Coleta de Dados

Esta seção tem como finalidade mostrar os detalhes metodológicos da coleta de dados realizada neste TCC.

2.5.1 População, amostra e sujeitos

O primeiro *survey* possui como características:

População: 43 pessoas responderam o questionário.

Amostra: Foi considerada toda a população como amostra.

Sujeitos: 43 pessoas que responderam ao questionário on-line divulgado em comunidades de CMSs na rede social facebook.

Já o segundo *survey* possui como características:

População: 80 pessoas responderam ao questionário.

Amostra: Foi considerada toda a população como amostra.

Sujeitos: 80 pessoas que responderam ao questionário on-line divulgado em comunidades de CMSs na rede social facebook e em listas de e-mail no Gmail relacionadas a CMS.

Em ambos os questionários não foram feitas perguntas para a classificação do sujeito, pois não era relevante para a pesquisa dados como sexo, idade e outros. Bastava apenas o sujeito ter conhecimento sobre CMS, trabalhar com CMS ou já ter desenvolvido aplicações web com auxílio de CMSs para responder as perguntas.

2.5.2 Instrumentos de Coleta de Dados

O primeiro questionário foi elaborado com base nas características de CMSs encontradas na literatura. Estas características estão descritas na Seção 6.1 e detalhadas no Apêndice B. Além disso, foi considerado também após o mapeamento, as características SQuaRE relacionadas. O relacionamento de características de CMS e características SQuaRE está detalhado na Tabela 25.

Devido a grande quantidade de características encontradas para a realização do questionário, as características foram agrupadas em características SQuaRE e foram separadas para as perguntas, aquelas mais citadas de acordo com a literatura.

A partir das considerações feitas, foi elaborado um questionário misto com 15 perguntas. Este questionário foi aplicado nas comunidades de Drupal, Wordpress, Joomla, Plone e desenvolvimento web na rede social facebook dos dias 03 aos dias 10 de maio de 2015.

Já o segundo questionário foi elaborado com base nas métricas definidas no Apêndice D, as quais foram baseadas nos resultados obtidos com a aplicação do Questionário 1.

O segundo questionário é composto por 19 perguntas fechadas, ambas constituídas por uma escala de importância de 4 pontos (1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante), sem o ponto neutro. Optou-se pela retirada do ponto neutro, porque não havia garantia de se conseguir uma amostra grande de entrevistados no período de tempo estipulado. Uma amostra grande seria maior ou

igual a 200 entrevistados.

O Questionário 2 foi aplicado na rede social Facebook em comunidades de Drupal, Wordpress, Joomla, Magento, Plone, desenvolvimento web, design responsivo e outros dos dias 28/05/2015 ao 12/06/2015. Além disso, o questionário também foi aplicado em listas de e-mail no Gmail durante os mesmos dias.

O detalhamento das perguntas de ambos os questionários está no Apêndice C.

2.5.3 Procedimentos de Coletas de dados

Os procedimentos adotados para a realização da coleta de dados para ambos os questionários foram:

1. Elaborar o questionário online com base nas características levantadas.
2. Divulgar questionário em comunidades de CMS e comunidades de desenvolvimento Web.
3. Coletar respostas dos entrevistados.
4. Analisar resultados.
5. Estabelecer conclusões.

2.6 Planejamento

A Figura 1 mostra uma EAP (Estrutura Analítica de Projeto) do trabalho de conclusão de curso. O tempo foi dividido em bimestres, intervalos de tempo de dois meses, e observe que os pacotes de trabalho possuem um índice S que significa semana, seguido de um índice numérico que significa o número da semana.

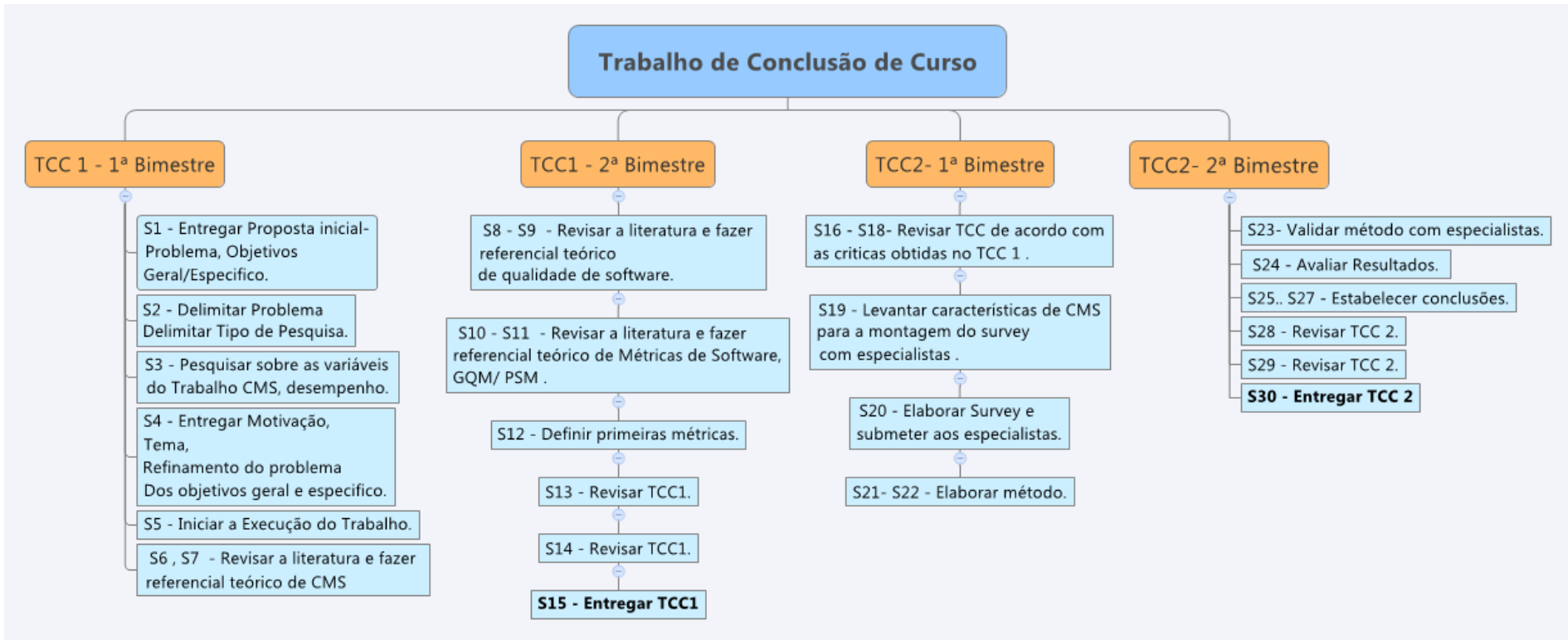


Figura 1 – Estrutura Analítica do Projeto de TCC.

2.7 Organização da Pesquisa

O desenvolvimento da pesquisa será organizado em capítulos que se seguem da seguinte forma.

- No Capítulo 3 serão abordados conceitos chaves para o estudo de CMSs. Alguns destes conceitos são: vantagens e desvantagens no uso de CMSs para o desenvolvimento web e os critérios de seleção para a escolha dos CMSs que serão objetos de estudo desta pesquisa.
- No Capítulo 4 serão abordados conceitos básicos de qualidade de software e conceitos referentes a série ISO/IEC 25000 de normas SQuaRE.
- No Capítulo 5 serão apresentados conceitos sobre métricas de software e métodos obtidos da literatura que são especializados na definição de medições, para o objetivo de avaliar as características que serão utilizadas na escolha de CMSs. Foram estudados os métodos: GQM (*Goal, Question, Metric*) e o PSM (*Practical Software Measurement*).
- No Capítulo 6 será apresentado o passo a passo para a construção do método proposto, que compreende desde o levantamento de características de CMS na literatura, a estruturação do método, até a validação por surveys das características levantadas e das métricas mapeadas.
- No Capítulo 7 serão apresentadas as conclusões obtidas com a realização do trabalho e sugestões de trabalhos futuros.

3 Content Management System

Este capítulo tem como objetivo apresentar conceitos básicos sobre *Content Management System*(CMS), que do inglês significa Sistemas Gerenciadores de Conteúdo. Nesta seção também será mostrado quais critérios serão utilizados para a escolha dos CMSs a serem avaliados futuramente no trabalho, além da revisão de literatura usada para a construção do capítulo.

3.1 Revisão de literatura para o capítulo

Na Seção 2.4.3 é descrito brevemente sobre os critérios de inclusão/exclusão de literatura usados neste TCC. Para este capítulo, a revisão de literatura abordou os seguintes critérios de inclusão:

1. Artigos que abordem as palavras **CMS**, ou ***Content Management System***, ou **Sistemas Gerenciadores de Conteúdo** ou quaisquer artigos que mencionem um CMS específico (*Wordpress, Joomla, Drupal, Alfresco, ModX*, entre outros) no *abstract* ou no título.
2. Teses, dissertações, monografias, relatórios técnicos, livros ou enciclopédias que abordem assuntos relacionados às mesmas palavras citadas no Item 1 desta seção.
3. Publicações em sites na web ou relacionadas, porém nestes casos a publicação em sites deverá estar apoiada por outra referência de natureza citadas no Item 1 desta seção.

Além disso, as bases de dados consultadas foram as mesmas definidas na Seção 2.4.1. Não foram definidas expressões de busca para este capítulo.

3.2 Definições Básicas

Um CMS é um software que mantém o controle da informação presente em uma determinada página web. Estas informações podem ser textos, fotos, músicas, vídeos, ou documentos (XIANG; YU, 2010a). Ele tem como objetivo organizar, favorecer a criação, administrar, distribuir, comunicar e disponibilizar informação em um dado web site (BARRÉRE; AVELINO; EDUARDO, S/D).

CMSs são uma boa solução para quem deseja organizar informações e principalmente criar e gerir conteúdos em vários contextos, dentre eles o contexto empresarial (BOUKAR; MUSLU, 2013).

3.2.1 CMSs e suas características

Sistemas gerenciadores de conteúdo são softwares e todo software possui características funcionais e não funcionais que atendem aos requisitos utilizados para suas concepções.

Por este motivo, apesar deste trabalho não versar especificamente sobre requisitos de software, a subseção seguinte faz uma breve explanação sobre os tipos de requisitos (funcionais e não funcionais) que caracterizam um CMS.

3.2.1.1 Características gerais (Requisitos) de CMSs

Para Filho (2003), requisitos são as características que delimitam os critérios de aceitação de um produto. O autor ainda acrescenta que, à medida que o processo de desenvolvimento está sendo executado, novos requisitos podem surgir, porém acrescentar características a um produto significa aumentar o custo do projeto e de sua produção.

Segundo LEITE (2001), requisitos de software são o conjunto de características que expressam as necessidades dos clientes e que, ao mesmo tempo, condicionam a qualidade do software.

De acordo com KOTONYA e SOMMERVILLE (1998), requisitos de software é a caracterização de serviços que o sistema deve prover, restrições que o sistema deve possuir e conhecimentos que serão precisos para construí-lo. Conhecendo esses elementos, a atividade de análise e negociação é facilitada, pois é possível identificar os envolvidos e também o ponto de partida de um requisito.

Os autores ainda afirmam que a Engenharia de Requisitos (ER) é a subárea da Engenharia de Software que trata do processo de definição dos requisitos de software. Esse processo é sistemático e atinge diversas atividades, tais como elicitação, análise e negociação, validação e gerência de requisitos.

Bortoli (1999) diz que requisitos funcionais definem as funções que o sistema ou componente do sistema, devem executar, ou seja, o comportamento do sistema ou de seus componentes que transforma entradas para produzir saídas.

Cysneiros (2001) diz que os requisitos funcionais definem funções ou serviços que um software deve ser capaz de oferecer ou de executar. Estas funções ou serviços na maioria das vezes são definidas como processos que utilizam de entradas para produzir saídas.

De acordo com [Cysneiros e Leite \(1997\)](#), os requisitos não funcionais, descrevem comportamentos e restrições que o software deve possuir.

[Bortoli \(1999\)](#) diz que requisitos não funcionais também são chamados de requisitos de qualidade. Esses requisitos descrevem tanto restrições no produto (desempenho, confiabilidade, interface de usuários, segurança) podendo, devido aos seus níveis de exigência, causar restrições no processo de desenvolvimento como aumento de custo e prazos de desenvolvimento.

Existem várias classificações para requisitos não funcionais propostas na literatura. As figuras 2 e 3 são algumas delas:

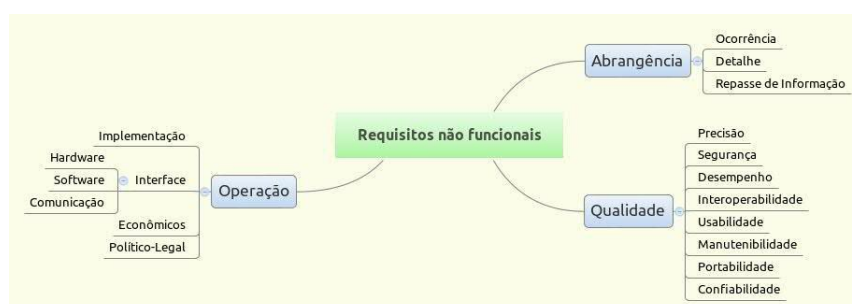


Figura 2 – Classificação de Requisitos Não Funcionais de ([MAMANI, 1999](#))

Já [SOMMERVILLE \(1992\)](#) apresenta uma classificação um pouco diferente. Para este autor, os requisitos não funcionais podem ser agrupados pelas seguintes Classes: Requisitos de Produto, Requisitos Externos e Requisitos de Processo.

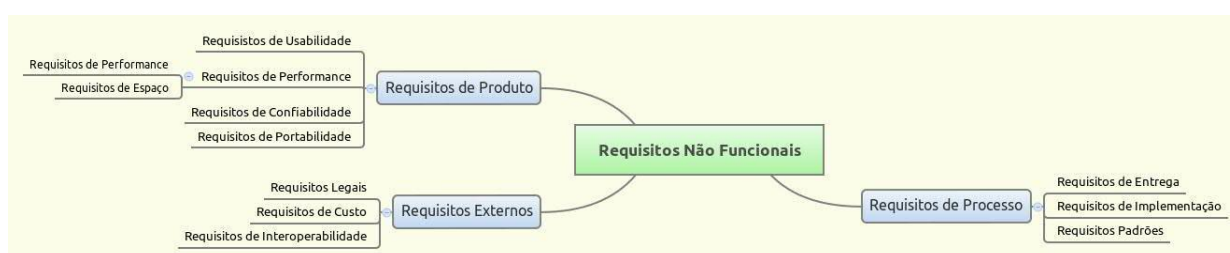


Figura 3 – Classificação de Requisitos não Funcionais por [SOMMERVILLE \(1992\)](#).

3.2.2 Vantagens no uso de CMS para o desenvolvimento web

O uso de CMSs no contexto de desenvolvimento web possui diversas vantagens. [Barrère, Avelino e Eduardo \(S/D\)](#), [Costa, Caminha e Lima \(2012\)](#), [Szybalski e Zabierowski \(2011\)](#) e [Santos \(2013\)](#) listam algumas delas:

- Uso de modelos pré definidos de páginas web (os chamados templates), que garantem a consistência de exibição do site como um todo.
- Funcionalidades podem ser incorporadas por meio de componentes pré construídos (os chamados plugins).

- Possibilidade de construir e administrar um website sem a necessidade de conhecer linguagens de programação para web.
- No contexto empresarial há redução de mão de obra com manutenção de sites por permitir maior desempenho no desenvolvimento.
- Permite gerir com facilidade o conteúdo do site separadamente do design.
- Basta ter um browser para criação e manutenção não sendo necessários nenhum software adicional.

3.2.3 Desvantagens no uso de CMS para o desenvolvimento web

Apesar de possuir inúmeras vantagens o uso de CMS ainda apresenta algumas desvantagens. [Barrère, Avelino e Eduardo \(S/D\)](#) e [Santos \(2013\)](#) listam as seguintes desvantagens, dentre outras:

- Apesar de trazer consistência a página web o uso de modelos pré-definidos também impõe restrições e limitações características do modelo a ser usado.
- A aparência do site fica limitada a quantidade de templates disponíveis para o CMS a ser usado.
- Apesar de criar sites com facilidade e evitar o desenvolvimento web desde do início, para se utilizar todos os recursos provenientes de um CMS é necessário um certo tempo de estudo para um bom entendimento e aproveitamento de suas funcionalidades.
- O backup da página é possível apenas no servidor.

3.3 Uso de CMS na indústria de Software

Devido às inúmeras vantagens de utilização, o uso de CMSs na indústria de software é bastante ampla. [Barrère, Avelino e Eduardo \(S/D\)](#) e [Santos \(2013\)](#) apresentam alguns exemplos de uso de CMS na indústria de software.

- Blogs.
- Sites para *e-commerce*.
- Sites para empresas ou instituições.
- Comunidades virtuais.
- Fóruns de discussão.

- Revistas e jornais online.
- Livros colaborativos.

3.4 Critérios para a escolha de CMSs e escolha de CMSs candidatos

No mercado de Software existem várias opções de CMSs disponíveis alguns muito populares e já consolidados, enquanto outros mais desconhecidos. Alguns CMSs são softwares livre enquanto outros proprietários.

O site [VisionWidget \(2014\)](#) apresenta uma visão rápida de vários produtos de CMS disponíveis no mercado. Dentre elas podem se destacar o Plone, Contao, Concrete5, Silverstripe, MODx que, embora não tão conhecidos, são citados pela ([VISIONWIDGET, 2014](#)).

O objetivo desta seção é justificar a escolha dos CMSs a serem alvos de estudo no trabalho. Os critérios investigados foram:

- **Licenciamento:** que para este trabalho foi investigado apenas o fato do CMS ser software livre ou proprietário, não se preocupando com detalhes relativos a licenças de software.
- **Popularidade:** investigado por meio do tamanho das comunidades em redes sociais e o número de *commits* nos repositórios Git Hub para os softwares livres.

3.4.1 Software Livre / Software Proprietário

A Engenharia de Software é bastante diferente se comparada com as demais engenharias. Uma diferença é que as pessoas que desenvolvem software podem publicá-los para que qualquer outra pessoa use, modifique ou construa trabalhos derivados com código aberto. Estes produtos de software são os chamados softwares livre ou software *open source* ([YOSHIURA, 2011](#)).

Softwares livres podem ser adaptados e modificados livremente para a resolução de um determinado problema ([AGUIAR et al., 2007](#)).

Para o software proprietário a cópia, redistribuição, ou modificação são proibidos pelo seu criador ou distribuidor e em geral apresentam alto custo restringindo a sua utilização ([AGUIAR et al., 2007](#)).

A Tabela 1 apresenta 15 soluções em CMS previamente mapeadas em software livre e software proprietário. Soluções citadas pelos autores [OpenSourceCMS \(2014\)](#), [VisionWidget \(2014\)](#), [MarketingdeConteúdo \(2014\)](#), [CmsMatrix \(2014\)](#), [CoreMedia \(2014\)](#), [Sherpa \(2014\)](#) e [Schwingel \(2009\)](#).

Tabela 1 – Mapeamento de CMSs em Software Livre e Software Proprietário

CMS	Software Livre	Software Proprietário
Joomla	x	
Drupal	x	
WordPress	x	
SilverStripe	x	
Plone	x	
Contao	x	
Modx	x	
Concrete5	x	
Pligg	x	
Vignette		x
CoreMedia		x
Milenium Cross Media		x
Notitia		x
Vinias		x
Sherpa		x

Segundo [Sherpa \(2014\)](#), valores de licenciamento de produtos de CMSs proprietários podem ser relevantes. A empresa fornecedora do Sherpa cita valores que podem chegar na ordem de grandeza de milhares de dólares por licença (10k - 75K). Assim sendo, a proposta do software livre de reduzir os custos de aquisição de licenças para o uso de produtos de software foi priorizada na escolha dos CMSs a serem estudados. De fato, [Karels \(2003\)](#) afirma que o uso de software livre está cada vez mais popular, pois apresenta várias vantagens, tanto do contexto de uso, quanto do contexto de desenvolvimento. Dentre algumas vantagens ([KARELS, 2003](#)):

- O custo de aquisição é baixo. Embora o software proprietário possua mais opções para suporte e treinamento, o seu custo de aquisição é maior se comparado a um software livre.
- O apoio comunitário no processo de desenvolvimento é frequente. Dependendo da comunidade uma dúvida pode ser respondida de forma rápida, como pode demorar várias semanas para ser respondida.
- O projeto pode ter muitos colaboradores voluntários, porém as habilidades e a disponibilidade desses colaboradores podem variar de forma significativa.
- Funcionalidades podem ser adicionadas com o projeto em andamento. Essa adição de funcionalidades é priorizada de acordo com as necessidades dos usuários do software

livre em questão.

Sendo assim, as opções mapeadas como software livre foram priorizadas nas análises da próxima seção, juntamente com o critério de avaliar suas respectivas popularidades.

3.4.2 Popularidade

O software livre vem se tornando bastante popular nos últimos anos. Essa popularidade se deve à internet, a qual permite uma distribuição rápida e em larga escala dos softwares que seguem esse modelo. Entretanto, a internet não colaborou apenas para a popularidade dos softwares livres, mas também permitiu atrair vários contribuidores bastando apenas o interesse dos mesmos (SALGADO; ATOJI, 2008).

Para o conceito popularidade, serão avaliados os nove CMSs Software Livre mapeados na Seção 3.4.1. Para justificar este conceito foram escolhidos três parâmetros, são eles:

- Número de contribuições – mensurada a partir do número de *commits* no repositório segundo a página das aplicações (GITHUB, 2014). Cabe ressaltar que foi preferido utilizar o número de *commits* no repositório de contribuições de melhorias do produto, independentemente do número de pessoas cadastradas como contribuintes, na suposição de que, apesar de um número menor de contribuintes, contribuições eram mais frequentes.
- Número de “opiniões favoráveis” na rede social Facebook - número de pessoas na maior comunidade do CMS na rede social Facebook, segundo pesquisa realizada no site (FACEBOOK, 2014).
- Número de “opiniões favoráveis” no Twitter” - número de pessoas na maior comunidade do CMS na rede social Twitter, segundo pesquisa realizada no site (TWITTER, 2014).

A partir das investigações feitas nos sites (FACEBOOK, 2014), (TWITTER, 2014) e (GITHUB, 2014) foram obtidos os dados da Tabela 2 ¹.

¹Dados atualizados até o dia 07/06/2015.

Tabela 2 – Popularidade dos CMSs Mapeados

CMS	Comunidade	Facebook	Twitter
Joomla	22.064	159.051	57.064
Drupal	18.840	79.161	58.932
WordPress	29.909	925.419	1.397.303
SilverStripe	14.364	3.374	3.373
Plone	71	1.173	3.582
Contao	5.347	2.855	1.940
Modx	7.858	3.349	4.593
Concrete5	13.353	2.268	5.718
Pligg	1.146	580	807

As Figuras 4, 5, 6, 7, 8, 9 mostram a visão gráfica da popularidade a partir da Tabela 2.

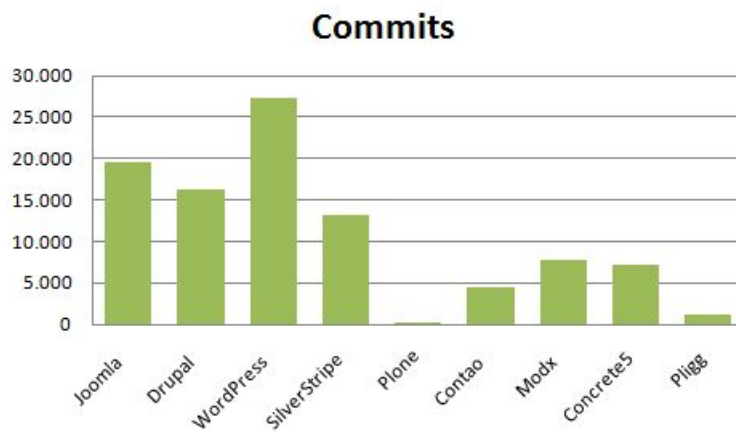


Figura 4 – Popularidade pelo número de commits no git hub.

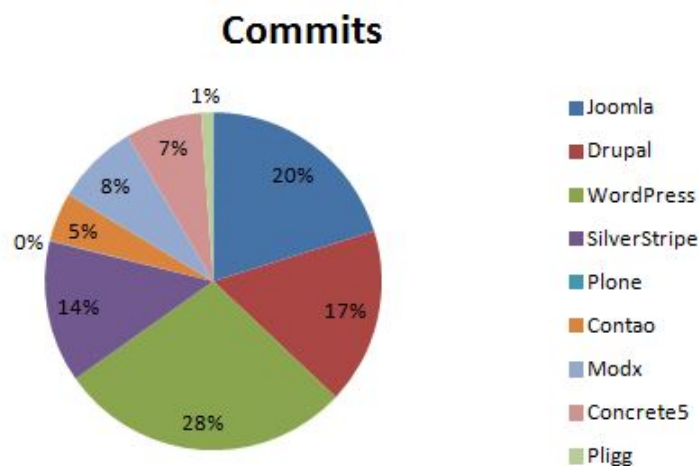


Figura 5 – Percentual de popularidade pelo número de commits no git hub.

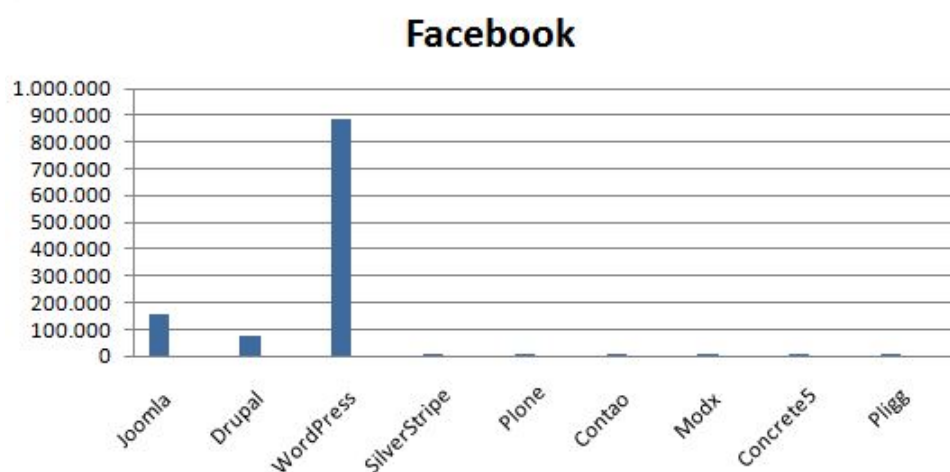


Figura 6 – Popularidade de acordo com a rede social Facebook.

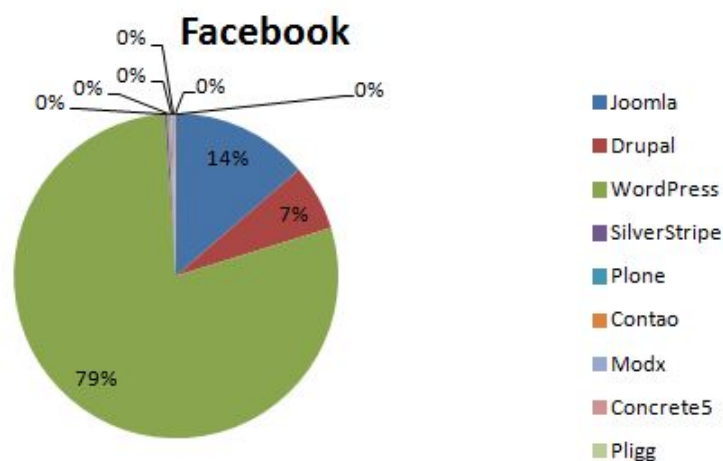


Figura 7 – Percentual de popularidade de acordo com a rede social Facebook.

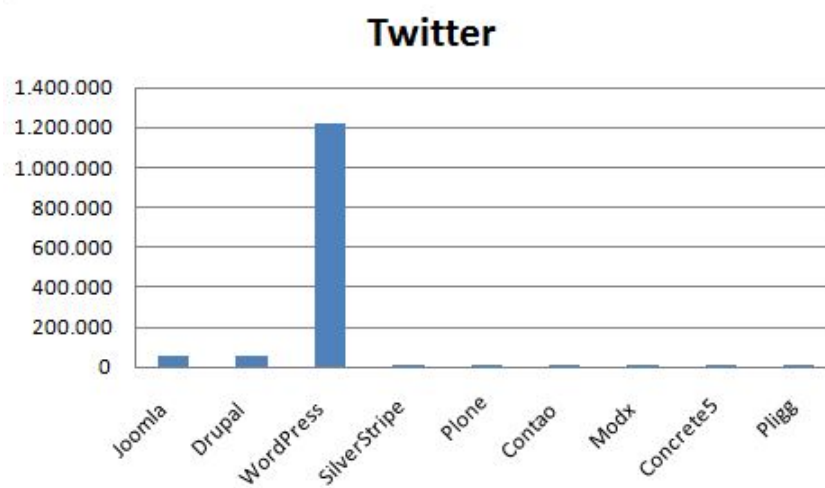


Figura 8 – Popularidade de acordo com a rede social Twitter.

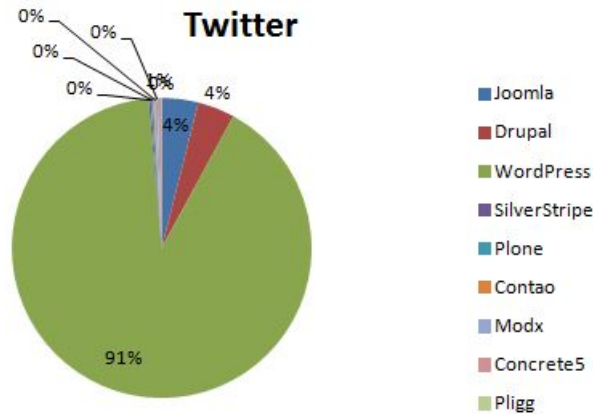


Figura 9 – Percentual de popularidade de acordo com a rede social Twitter.

A partir das Figuras 4, 5, 6, 7, 8, 9 e da Tabela 2 foi constatado que os CMSs mais populares são o Wordpress, Joomla, e Drupal.

Além dos parâmetros observados outras pesquisas de sites da internet apontam os CMS mais populares.

Para a [MarketingdeConteúdo \(2014\)](#), o WordPress é o CMS mais usado no mundo. Ele tem licença livre, e pode ser instalado com facilidade, a empresa gosta de mencionar a sua rápida instalação de 5 minutos. Apesar de ser considerado uma plataforma feita para blogs, o WordPress é bastante mutável e pode ser usado na construção de páginas corporativas, e-commerce, jornais e demais itens mencionados na seção 3.3 deste trabalho.

Logo depois do WordPress aparece o Joomla como segunda ferramenta de CMS mais usada. O debate sobre qual ferramenta é a melhor já existe há muito tempo, com defensores de ambos os lados. Apesar da competição o Joomla ainda perde em quesitos como simplicidade e funcionalidades, não sendo aconselhado para pessoas iniciantes no desenvolvimento web, segundo o site [MarketingdeConteúdo \(2014\)](#).

O Drupal mesmo apresentando um crescimento no número de usuários, ainda é pequeno se comparado com WordPress ou Joomla. O sistema também é mais complexo do que seus concorrentes, não sendo indicado para usuários leigos, apenas para aqueles que já possuem alguma experiência com desenvolvimento, segundo o site [MarketingdeConteúdo \(2014\)](#).

A [OpenSourceCMS \(2014\)](#), utilizou um complemento de navegador chamado *Wapalyzer*, que tem por finalidade descobrir quais tecnologias foram usadas para a construção de tais sites. Dentre essas tecnologias que podem ser encontradas, podem-se destacar CMS, frameworks e ferramentas de análise de código. Com o auxílio desta ferramenta a OpenSource CMS, descobriu que o Wordpress, o Joomla e o Drupal são os mais utilizados na composição dos sites avaliados respectivamente, tomando como exemplo uma amostra de 1 % de sites da internet.

Com base nas pesquisas efetuadas desta seção foram escolhidos para fazerem parte do estudo como CMSs populares: Joomla, Drupal e Wordpress.

3.5 Vantagens e Desvantagens dos CMS mais populares

Esta seção tem como objetivo analisar algumas vantagens e desvantagens dos CMSs escolhidos Joomla, WordPress e Drupal, porém antes de mostrar as vantagens e desvantagens de tais CMSs será apresentada uma visão rápida de cada CMS.

1. **Wordpress:** É um software livre que possui licença GPLv2². Ele tem como principal objetivo a criação de blogs e fornece ao seu usuário vários recursos para a criação destes blogs. Dentre esses recursos, podem ser destacados os painéis de administração, que são uma opção para definir o comportamento e a apresentação do site (REIS et al., 2011).
2. **Drupal:** Também é um software livre que possui licença GPLv2. Trabalha com alguns conceitos chaves. Como **Módulos**, que são os conjuntos de códigos que são responsáveis por prover as principais funcionalidades do Drupal; a **Taxonomia**, que é o sistema de classificação de conteúdo; o **Tema**, que faz o gerenciamento do que é exibido no site controlando o layout, cores e aspecto gráfico (REIS et al., 2011).
3. **Joomla:** Distribuído sob licença GPLv2 é um CMS software livre que permite total customização do layout, tornando possível desde a simplificação de buscas até o suporte à múltiplas línguas(internacionalização)(REIS et al., 2011).

Costa, Caminha e Lima (2012), Reis et al. (2011) elencam em seus trabalhos várias vantagens e desvantagens dos CMSs Joomla, Wordpress e Drupal. A relação de vantagens e desvantagens de cada CMS está na Tabela 3.

²A *General Public License* é uma licença projetada por Stalman para software livre possui quatro princípios básicos. Dentre esses princípios destacam-se A liberdade para executar, distribuir, desenvolver melhorias para o programa e produzir trabalhos derivados, desde que o software derivado, ou seus trabalhos derivados sejam também livres. (SMITH, 2007)

Tabela 3 – Vantagens e Desvantagens dos CMSs, segundo (COSTA; CAMINHA; LIMA, 2012), (REIS et al., 2011)

CMS	Vantagens	Desvantagens
Joomla	Fácil de instalar. Boa usabilidade. Software Livre. Flexibilidade e interatividade alta. Forte documentação. Comunidade bastante ativa.	Taxonomia Limitada. Boa parte de seus plugins e templates são pagos.
Wordpress	Fácil de instalar. Boa usabilidade. Software Livre. Flexibilidade e interatividade alta. Bom para a construção de blogs. Total conformidade com os padrões da World Wide Web Consortium.	Taxonomia Limitada. Segurança limitada. Performance Limitada. Mecanismo de busca não é preciso.
Drupal	Boa taxonomia. Software Livre. Possui mais recursos em nível de desenvolvimento. Fácil configuração para interagir com outros sites e tecnologias. Liberdade para a alteração de um template específico.	Curva de aprendizado alta. Usabilidade ruim. Difícil de instalar. Inadequado para usuários com pouca experiência no uso de CMS.

A partir da Tabela 3 foi constatado que os CMSs possuem vantagens em comuns como, a facilidade de instalar e o fato de serem open source. Entretanto mesmo

oferecendo várias vantagens os CMSs ainda apresentam limitações quanto a Taxonomia (Joomla e Wordpress) e curva de aprendizado e usabilidade (Drupal).

3.6 Comparativos já Realizados

Patel, Rathod e Parikh (2011) propuseram uma pesquisa que buscava avaliar três CMSs Joomla, Wordpress e Drupal com o objetivo de aferir o desempenho desses CMSs observando algumas características. A pesquisa foi composta de dois estudos de caso, em um deles as avaliações eram feitas em um servidor local. Para o servidor local, as características observadas são detalhadas na tabela 4:

Tabela 4 – Características avaliadas pelo experimento de Patel, Rathod e Parikh (2011)

Característica	Descrição
Tempo de carregamento da página	Tempo em Milissegundos (ms).
Tamanho da página	Tamanho total da página em (KB).
Total de Requisições	Número de pedidos enviados ao servidor para se carregar uma página.
Total de Arquivos CSS	Número de arquivos CSS usados pelo CMS para fazer uma página.
Total de arquivos JS	Número de arquivos Java Script usados pelo CMS para fazer uma página.
PLT	Tempo em que se armazena o conteúdo na memória cache antes da página carregar.
PS	Quantidade de dados armazenados na memória cache antes da página carregar.

Depois de estabelecidas as características, Patel, Rathod e Parikh (2011) estabeleceram três experimentos dentro do primeiro cenário do servidor local:

- 1^a Experimento - Página comum sem plugins.
- 2^a Experimento - Plugin de calendário e informação textual.
- 3^a Experimento - Calendário, informação textual, imagem, plugin de relógio.

Após os dados levantados, Patel, Rathod e Parikh (2011), obtiveram o ranking mostrado na Tabela 5 e nas Figuras 10, 11, 12.

Tabela 5 – Ranking gerado a partir do experimento de Patel, Rathod e Parikh (2011).

CMS/ Experimento	Wordpress	Joomla	Drupal
1ª Experimento	2ª	3ª	1ª
2ª Experimento	3ª	2ª	1ª
3ª Experimento	3ª	1ª	2ª

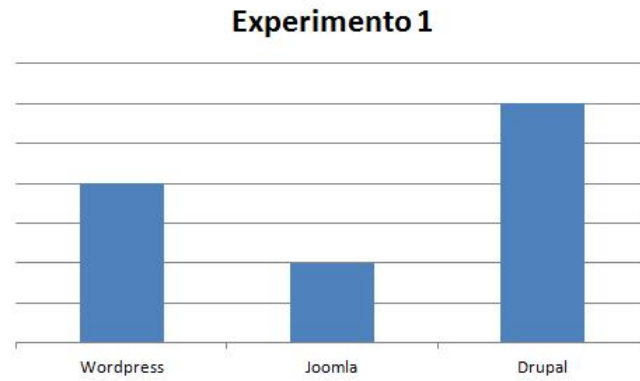


Figura 10 – Ranking de melhor desempenho no experimento 1

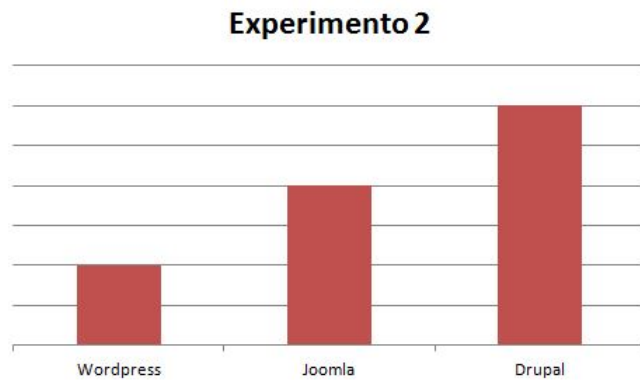


Figura 11 – Ranking de melhor desempenho no experimento 2

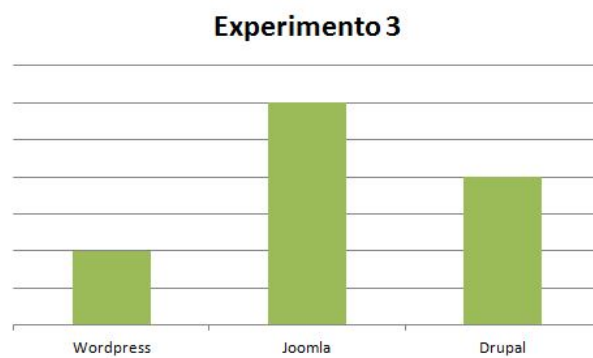


Figura 12 – Ranking de melhor desempenho no experimento 3

As Figuras 10, 11, 12 e a Tabela 5 demonstram que Patel, Rathod e Parikh (2011) obtiveram a conclusão que para um servidor local e os experimentos propostos o CMS Drupal obteve o melhor desempenho e o WordPress o pior.

Conforme visto, características carecem de definições mais objetivas para poderem ser considerados como bem definidos. Uma forma de objetivamente quantificar grandezas consiste em utilizar a teoria de métricas de software. A ISO/IEC-25000 (2005) propõe um conjunto de métricas para avaliar grandezas ligadas ao conceito de qualidade em software o que, por sua vez, será o foco de estudo dos próximos capítulos para o objetivo de definir um método baseado em características de CMS. Um método que poderá ser utilizado como insumo para a escolha de um determinado CMS em uma organização.

4 Qualidade de Software

Hamid e Hasan (2010) dizem que medir a qualidade do software na indústria não é uma tarefa fácil e a maioria das empresas não costumam implementar este conceito. Existem várias razões para que a qualidade não seja medida, dentre alguns dos motivos mais comuns estão os gestores que não sabem como medir, dificuldades para criar a infraestrutura necessária para a medição e o medo do resultado (KAN, 2002 apud HAMID; HASAN, 2010).

Durante muito tempo no contexto de desenvolvimento de software no Brasil, o uso da NBR-ISO/IEC-9126 (2003), apesar de não atingir a maioria das empresas foi maior dentre todas as normas utilizadas por empresas de software que desejassem alcançar a qualidade de seus produtos (MARINHO; SOUZA, 2010). Os autores afirmam que 5,7 % de 343 organizações pesquisadas utilizam a NBR-ISO/IEC-9126 (2003) como padrão para se alcançar a qualidade de software.

Porém a norma ISO 9126 foi substituída pela série de normas 25000, também chamadas de *Software Quality Requirements and Evaluation* (SQuaRE). As Normas SQuaRE trazem conceitos comuns com a antiga ISO 9126, além de novos conceitos apresentados a partir da Seção 4.2 (ISO/IEC-25010, 2011).

4.1 Revisão de literatura para o capítulo

Para este capítulo a revisão de literatura abordou os seguintes critérios de inclusão:

1. Artigos que abordassem as palavras **Qualidade de Software**, ou *Software Quality*, ou **SQuaRE**, ou relacionadas no seu *abstract* ou no seu título.
2. Teses, dissertações, monografias, relatórios técnicos, livros ou enciclopédias que abordassem assuntos relacionados as mesmas palavras citadas no item 1 desta seção.
3. Publicações em sites na web ou relacionadas, porém nestes casos a publicação em sites foi apoiada por outra referência de natureza citadas no Tópico 1 desta seção.

As bases de dados consultadas foram as mesmas definidas na Seção 2.4.1.

4.2 As Normas SQuaRE

A maior motivação para a criação das Normas SQuaRE foi a necessidade de se construir um conjunto harmônico de documentos, visto que faltava clareza na utilização

das normas de qualidade de produto. A SQuaRE visa obter uma série logicamente organizada e unificada com abrangência de dois processos principais: especificação de requisitos de qualidade e avaliação da qualidade de software, apoiados por um processo de medição (ISO/IEC-25000, 2005).

De acordo com a ISO/IEC-25000 (2005), os principais benefícios da série SQuaRE em Relação aos modelos anteriores são:

- Coordenar a orientação sobre a mensuração da qualidade dos produtos de software e avaliação da qualidade de software.
- Oferecer uma orientação para a especificação de requisitos de qualidade do produto de software.
- Harmonizar com a norma ISO/IEC 15939- Qualidade do processo de software ISO/IEC-15939 (2007), sob a forma de produto de Software de Qualidade de medição de referência.

4.3 Organização das normas Square

A Figura 13 a seguir mostra a organização das normas SQuaRE:

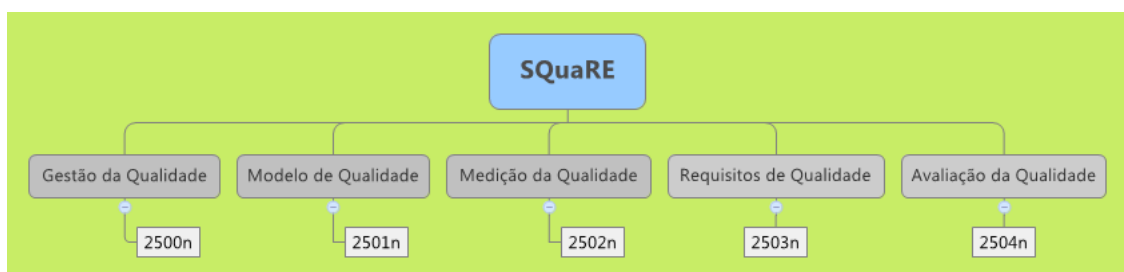


Figura 13 – Organização da série SQuaRE (ISO/IEC-25000, 2005)

De acordo com a Figura 13 as normas SQuaRE possuem como extensão:

1. **ISO/IEC 2500n – Divisão Gestão da Qualidade:** essa divisão determina os modelos, termos e definições a respeito das demais normas da série Square.
2. **ISO/IEC 2501n – Divisão Modelo de Qualidade:** essa divisão apresenta um modelo de qualidade detalhado, incluindo características de qualidade do produto e qualidade em uso. Além disso, as características de qualidade de produto são quebradas em subcaracterísticas. Essa divisão também oferece orientação a respeito do uso do modelo de qualidade.
3. **ISO/IEC 2502n – Divisão Medição da Qualidade:** essa divisão apresenta um modelo de referência para a medição de produtos de software, definições matemáticas

das medidas de qualidade e orientações práticas para a sua aplicação. As medidas apresentadas aplicam-se ao contexto de qualidade em uso e qualidade do produto.

4. **ISO/IEC 2503n – Divisão Requisitos de Qualidade:** essa divisão especifica o que são requisitos de qualidade. Estes requisitos podem ser usados no processo de elicitação de um produto de software a ser desenvolvido ou como entrada para um processo de avaliação.
5. **ISO/IEC 2504n – Divisão Avaliação da Qualidade:** essa divisão fornece requisitos, recomendações e orientações para a avaliação de produtos de software quando realizado por avaliadores, adquirentes ou desenvolvedores.

4.4 Qualidade de Software segundo a SQuaRE

Segundo a [ISO/IEC-25010 \(2011\)](#), a qualidade de um sistema é o grau em que o sistema satisfaz as necessidades explícitas e implícitas de seus usuários e, dessa forma, agrega valor. Estas necessidades explícitas e implícitas são representadas na série SQuaRE de normas de modelos de qualidade que categorizam a qualidade do produto em características que, em alguns casos, são ainda subdivididas em subcaracterísticas.

Esta decomposição hierárquica fornece uma análise conveniente de qualidade do produto, pois as características de qualidade definidas abrangem todos os aspectos de qualidade que são relevantes para a maioria dos produtos de software. Dessa forma, eles podem ser usados como uma lista de verificação para garantir a abrangência da qualidade ([ISO/IEC-25010, 2011](#)).

Para se chegar às medidas de características de qualidade ou subcaracterísticas, é necessário identificar um conjunto de propriedades que juntos atendem a característica ou subcaracterística. Após a identificação das propriedades é necessário identificar medidas para cada uma das propriedades encontradas. A partir da identificação das medidas das propriedades pode-se chegar a medidas derivadas que sirvam de insumos para medição de características e de subcaracterísticas ([ISO/IEC-25010, 2011](#)).

A [ISO/IEC-25010 \(2011\)](#) classifica a qualidade do software em características que estão subdivididas em subcaracterísticas de qualidade, conforme ilustra a Figura 14 apresentada a seguir:

Além disso, o modelo de qualidade é dividido em modelo de qualidade de uso e modelo de qualidade do produto ([ISO/IEC-25000, 2005](#)).

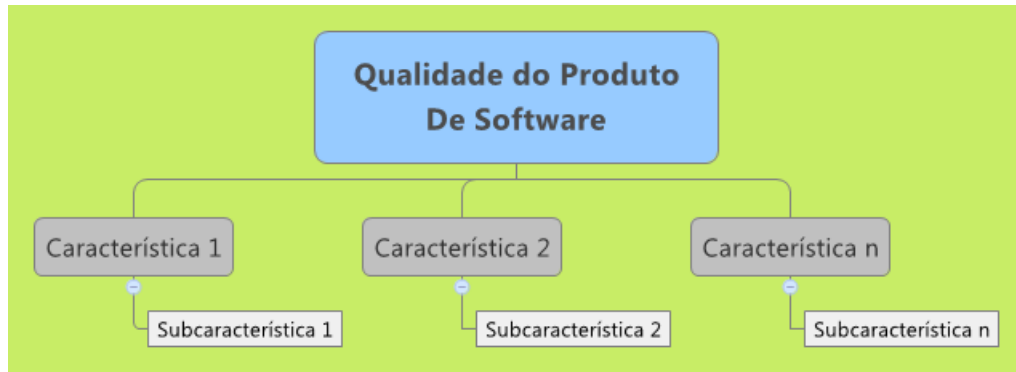


Figura 14 – Estrutura do modelo de qualidade da SQuaRE (ISO/IEC-25010, 2011)

4.4.1 O Modelo de Qualidade em Uso da Square

A qualidade em uso define cinco características relacionadas à evolução da interação com um sistema. Essas características, segundo a ISO/IEC-25010 (2011) são:

1. **Efetividade:** é a característica que diz respeito a capacidade que o software possui de atender a metas específicas sob condições particulares de uso levando em consideração a exatidão e a integridade.
2. **Eficiência:** é a característica que diz respeito a capacidade que o software possui de apresentar recursos que foram gastos ao atingir metas específicas sob condições particulares de uso levando em consideração a exatidão e a integridade.
3. **Satisfação:** é a característica que diz respeito a capacidade que o software possui de agradar seus clientes em um contexto de uso específico.
4. **Inexistência de risco:** é a característica que diz respeito a capacidade que o software possui de minimizar riscos econômicos, humanos, para a vida humana e ambientais em um contexto de uso específico.
5. **Cobertura de Contexto em Uso:** é a característica que diz respeito a capacidade que o software tem de possuir eficácia, eficiência inexistência de riscos e satisfação do cliente diante de um contexto de uso específico.

Essas características ainda podem ser quebradas em outras subcaracterísticas e essas podem ser mensuradas por meio de métricas de qualidade em uso ISO/IEC-25010 (2011), conforme mostra a Figura 15.

4.4.2 O Modelo de Qualidade do Produto segundo a norma Square

O modelo de qualidade do produto caracteriza um sistema ou software em oito características. Essas características segundo a ISO/IEC-25010 (2011) são:

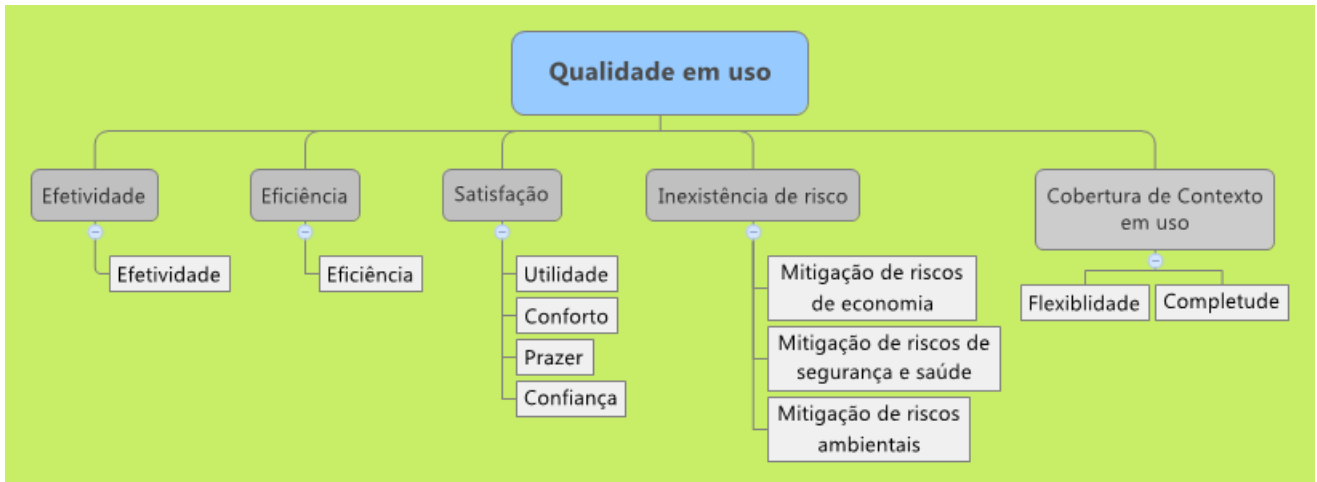


Figura 15 – Qualidade em Uso segundo a SQuaRE (ISO/IEC-25010, 2011)

- **Adequação Funcional:** característica que diz que um produto ou sistema deve fornecer funções que correspondam às necessidades explícitas e implícitas, quando usado sob condições especificadas.
- **Eficiência de Desempenho:** característica que estabelece o desempenho de um determinado sistema em relação à quantidade dos recursos utilizados sob condições estabelecidas.
- **Compatibilidade:** característica que estabelece que um produto, sistema ou componente deve trocar informações e / ou realizar suas funções necessárias, ao compartilhar o mesmo ambiente de hardware ou software.
- **Usabilidade:** característica que estabelece que um produto ou sistema deve ser usado por um usuário específico para o alcance de metas específicas com eficácia, eficiência e satisfação em um contexto de uso determinado.
- **Confiabilidade:** característica que diz que um sistema, produto ou componente executa funções específicas sob condições determinadas em um dado período de tempo.
- **Segurança:** característica que diz que um sistema ou produto protege as informações e dados, de modo que as pessoas, outros produtos ou sistemas possuam o grau de acesso de dados apropriado para os seus tipos e níveis de autorização.
- **Manutenibilidade:** característica que diz que um sistema possui a capacidade de ser modificado com determinado grau de eficácia e eficiência por um conjunto de mantenedores.
- **Portabilidade:** característica que diz que um sistema, produto ou componente pode ser transferido a partir de um hardware, software ou outro ambiente operacional com determinado grau de eficácia e eficiência.

Essas características ainda podem ser quebradas em outras subcaracterísticas e essas podem ser mensuradas por meio de métricas de qualidade do produto [ISO/IEC-25010 \(2011\)](#), conforme mostra a [Figura 16](#).

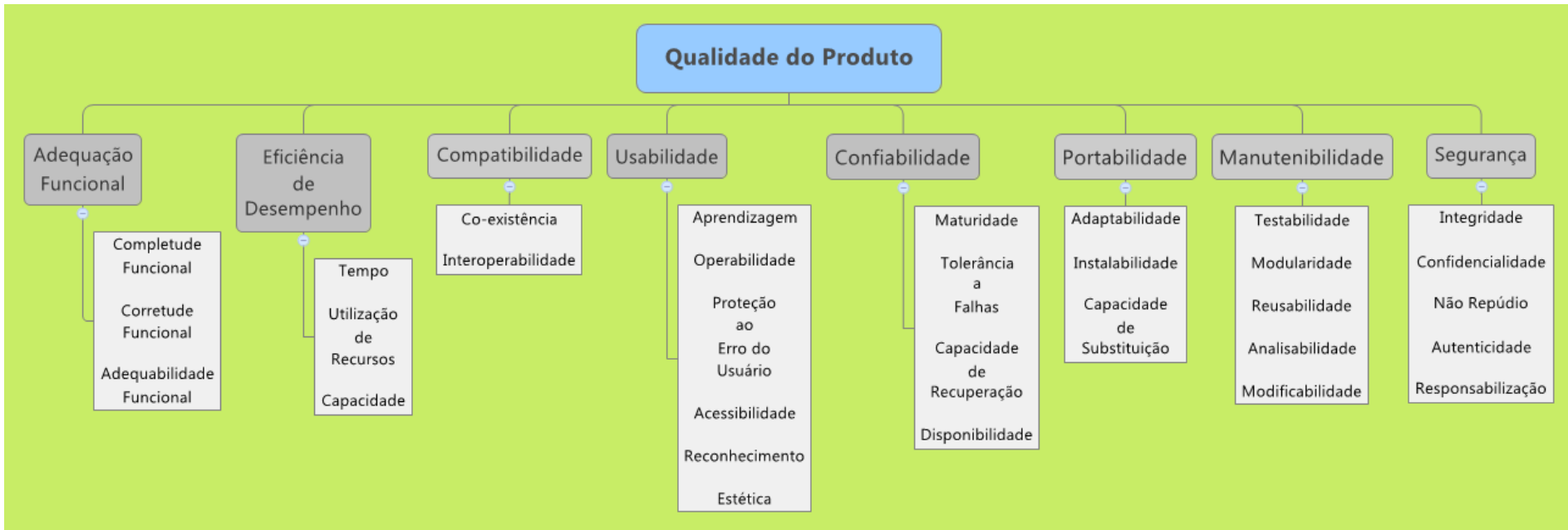


Figura 16 – Qualidade do Produto segundo a SQuaRE (ISO/IEC-25010, 2011)

4.4.3 Comparação entre a SQuaRE e a ISO / IEC 9126

A SQuaRE revisa a [ISO/IEC-9126-1 \(2001\)](#), e mantém alguns conceitos de qualidade de software enquanto acrescenta outras características com algumas alterações.

A tabela 6 mostra um comparativo entre o que existe na SQuaRE e o que existia na antiga ISO 9126 de forma equivalente.

Tabela 6 – SQuaRE X ISO 9126 ([ISO/IEC-25010, 2011](#))

#	Tem na Square	Como era na ISO 9126	Notas de Mudança
1	Qualidade em uso	Qualidade em uso	Manteve o nome, porém uma subcaracterística foi acrescentada e houve mudanças nas subcaracterísticas que já existiam.
1.1	Efetividade	Efetividade	Não houve alterações.
1.2	Eficiência	Produtividade	Nome alinhado com eficiência em ISO / IEC 25062 e ISO 9241-11.
1.3	Inexistência de Risco	Segurança	O conceito de segurança em nível de uso foi mudado e passou a abranger a capacidade que o sistema tem de lidar com riscos. Além disso foram acrescentadas subcaracterísticas.
1.4	Cobertura do Contexto em Uso	Não existia	Foram adicionadas na qualidade em uso como subcaracterísticas completude e flexibilidade.
1.5	Satisfação	Satisfação	Não existiam subcaracterísticas.
2	Qualidade do Produto	Qualidade externa/ Qualidade interna	Ambos os conceitos de qualidade interna e externa foram extintos e combinados em um único conceito chamado de qualidade do produto.
2.1	Adequação Funcional	Funcionalidade	Além de ter sido renomeada para adequação funcional para evitar confusão com outros sentidos de funcionalidade as subcaracterísticas que existiam na ISO 9126 foram substituídas por completude, corretude e adequação funcional.

2.2	Eficiência de Desempenho	Eficiencia	Foi renomeado para evitar conflito com a definição de eficiência na ISO/IEC 25062, além disso, foi adicionada uma nova subcaracterística chamada capacidade.
2.3	Compatibilidade	Não existia	Nova característica. Inclui como subcaracterísticas interoperabilidade e co-existência.
2.4	Usabilidade	Usabilidade.	Algumas subcaracterísticas foram renomeadas. A subcaracterística Acessibilidade foi acrescentada.
2.5	Confiabilidade	Confiabilidade	Foi acrescentada a subcaracterística de disponibilidade.
2.6	Segurança	Subcaracterística de Funcionalidade	Foi adicionada como uma característica, em vez de uma subcaracterística de funcionalidade, com subcaracterísticas confidencialidade, integridade, não-repúdio, prestação de contas e autenticidade.
2.7	Manutenibilidade	Manutenibilidade	Foram adicionadas duas novas subcaracterísticas a reusabilidade e a modularidade.
2.8	Portabilidade	Portabilidade	A subcaracterística co-existência foi movida para a característica de compatibilidade.

Os modelos de qualidade de software utilizam-se de medições. A medição é vital para a compreensão, controle e melhoria na engenharia (TARHAN; YILMAZ, 2014). O capítulo 5 define conceitos básicos de medição, medidas de software, e métodos usados para a definição de métricas.

5 Métricas de Software

Medição é o processo no qual são atribuídos números e símbolos as características dos itens do mundo real de modo que seja possível descrever com clareza esses atributos (FENTON; PFLEGEER, 1997, p. 5).

O resultado numérico é chamado de medida e este resultado pode ser aplicado tanto ao processo de desenvolvimento de software, quanto a um produto de software (SOLLINGEN; BERGHOUT, 1999). Além disso, uma medida pode ser considerada como uma indicação quantitativa da extensão, dimensão, tamanho ou capacidade de um determinado atributo de um processo ou produto (PRESSMAN, 2001, p. 81).

Este capítulo tem como objetivo apresentar uma breve noção sobre tipos de métricas de software e métodos para a definição de métricas.

5.1 Revisão de literatura para o capítulo

A Seção 2.4.3 apresenta critérios de inclusão e exclusão de literatura usados neste TCC. A revisão de literatura deste capítulo considerou os seguintes critérios de inclusão:

1. Artigos que abordem as palavras **Métricas de Software**, ou **GQM**, ou **PSM** ou **Software Metrics** no seu *abstract* ou no seu título.
2. Teses, dissertações, monografias, relatórios técnicos, livros ou enciclopédias que abordem assuntos relacionados as mesmas palavras citadas no item 1 desta seção.
3. Publicações em sites na web ou relacionadas, porém nestes casos a publicação em sites deverá estar apoiada por outra referência de natureza citadas no tópico 1 desta seção.

As bases de dados consultadas foram as mesmas definidas na Seção 2.4.1. Não foram definidas strings de pesquisa para este capítulo.

5.2 Tipos de Métricas de Software

Para um processo de desenvolvimento de software ou um produto ser medido é necessário o uso de métricas. Em um processo de medição podem ser usadas várias métricas ao mesmo tempo e essas mesmas métricas podem se repetir várias vezes (SOLLINGEN; BERGHOUT, 1999). Durante a medição de um produto de software, as métricas podem ser mapeadas de diversas formas (FENTON; PFLEGEER, 1997, pp 45-46).

Várias classificações de métricas de software podem ser encontradas na literatura, algumas delas são descritas por [Fenton e Pflieger \(1997, pp. 37-42, 60, 74-83\)](#), [Moller e Paulish \(1993\)](#), [Park, Goethert e Florac \(1996, p .9\)](#) e apresentados na Tabela 7:

Tabela 7 – Tipos de métricas de Software.

Classificação	Finalidade	Exemplo
Métricas de Produto	Medir atributos de produtos, documentos e artefatos oriundos do processo de desenvolvimento de software.	Tamanho do produto de software.
Métricas de Processo	Medir atividades do processo de desenvolvimento de software.	Densidade de defeitos de testes.
Métricas de Recurso	Medir atributos que são usados durante a execução do processo de desenvolvimento de software.	Tamanho da equipe.
Métricas Objetivas	Pode ser quantificada por meio de expressões numéricas ou gráficos.	Tempo de execução.
Métricas Subjetivas	Necessitam de avaliação pessoal.	Nível de satisfação do cliente, definida em classes como: baixa, média e alta.
Métricas Diretas	Mede interações em apenas uma única dimensão.	Número de funcionalidades, Esforço.
Métricas Indiretas	Mede interações entre mais de uma dimensão.	Produtividade.

Além das classificações sobre tipos de medições, outro conceito importante é o relativo às escalas de medição.

As escalas de medição definem várias maneiras em que uma métrica pode ser mapeada por meio de relações numéricas e empíricas. O uso de escalas de medição ajuda a entender o comportamento de entidades e definir valores para seus atributos ([FENTON; PFLEGEER, 1997, pp 45-46](#)).A Tabela 8 apresenta as escalas de métricas mais usadas na Engenharia de Software.

Tabela 8 – Tipos de escalas de métricas de Software Fonte: (PARK; GOETHERT; FLOORAC, 1996, p. 9); (FENTON; PFLEGEER, 1997, pp. 45-53)

Tipo	Descrição	Exemplos
Nominal	Um nome ou rótulo é atribuído como classe de valor do atributo, não existindo noção de ordem entre elas. Qualquer sistema simbólico ou representação é uma medida aceitável, mas não existe noção de magnitude associada.	Cor dos olhos de pessoas, classes de defeitos, nomes de linguagens, classes de custos (custos diretos, custos indiretos), etc.
Ordinal	Semelhante à escala anterior, mas acrescenta a noção de ordem entre os tipos estabelecidos. Números, se utilizados, significam apenas classificações e não é possível efetuar operações matemáticas com eles.	Os níveis de maturidade do CMM (níveis de 1 a 5); Complexidade de uma função de um sistema (baixa, média, alta).
Intervalar	Preserva a importância da ordem dos resultados da escala ordinal, e ainda possui informações sobre o tamanho dos intervalos que separam seus pontos. Permite realizar adições e subtrações, mas não permite multiplicações e divisões.	Temperatura, intervalos de: datas, horários, etc.
Racional	Preserva a ordem, o tamanho dos intervalos entre entidades, mas apresenta também as razões entre elas. Incorpora o elemento zero absoluto (representando a total falta do atributo). Todas as funções aritméticas podem ser utilizadas aplicadas em cada intervalo do mapeamento, gerando resultados significativos.	Tamanho, peso, altura, tempo entre falhas, valores de custos, prazos, esforços, etc.
Absoluta	Consiste em um tipo especial de escala racional, na qual somente são admissíveis multiplicadores unitários, isto é, a medição é realizada por meio da contagem do número (quantidade) de elementos de uma determinada entidade.	Número de defeitos encontrados no software, número de pessoas trabalhando em uma equipe, número de ocorrências de um determinado tipo, etc.

Na Engenharia de Software existem abordagens que auxiliam na definição das métricas, tanto de produtos, quanto de processos de software e recursos. Dois exemplos de métodos bem conhecidos são o *Goal Question Metric* (GQM) e o *Practical Software Measurement*(PSM).

5.3 GQM

O método *Goal, Question, Metric* (GQM) é uma abordagem utilizada para a medição composta por quatro fases(SOLLINGEN; BERGHOUT, 1999), conforme mostra a figura 17.

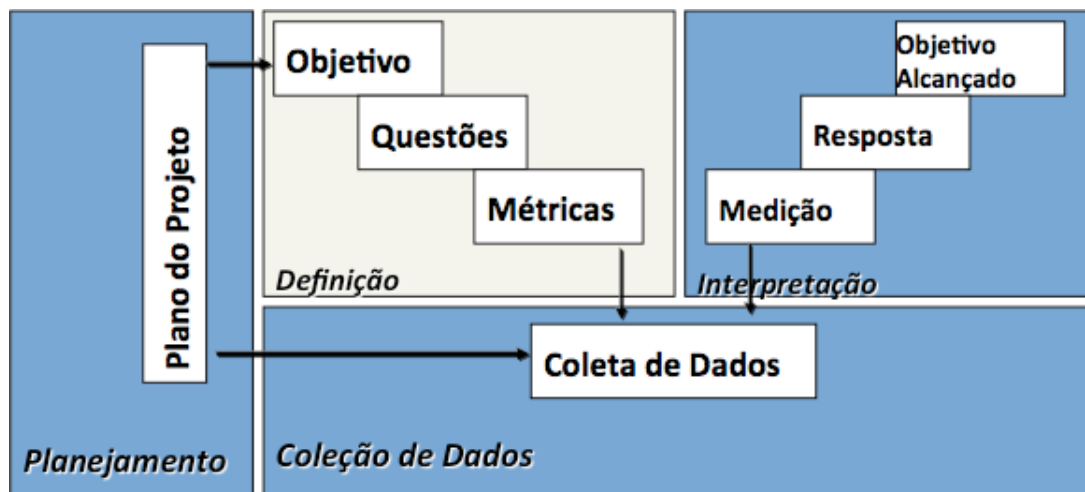


Figura 17 – Fases do GQM (SOLLINGEN; BERGHOUT, 1999)

A Figura 17 apresenta quatro fases, sendo elas: planejamento, definição, coleta de dados e interpretação. Cada uma delas é explicada a seguir.

1. **Fase do Planejamento:** nesta fase o planejamento da definição das medições é feito, com atividades características como: escolher o(s) objetos(s) a serem mensurados, com suas características para que sejam avaliadas por meio de medições. Além disso, a equipe que trabalhará no projeto de definições será formada, assim como todas as etapas do trabalho serão configuradas (CARD, 2003).
2. **Fase de Definição:** nesta fase a estratégia de medição é estabelecida. São definidos os objetivos, as questões, as métricas e as hipóteses e essas definições são documentadas (CARD, 2003).
3. **Fase de Coleta de Dados:** nesta fase a coleta de dados ocorre resultando em um levantamento de dados (CARD, 2003).

4. **Fase da Interpretação:** nesta fase os dados são processados de acordo com as métricas definidas para a medição dos resultados, para que as respostas das questões sejam encontradas e para que os objetivos possam ser avaliados (CARD, 2003).

Para Basili, Caldiera e Rombach (1994), o GQM pode ser dividido em três níveis principais, conforme ilustra a Figura 18:

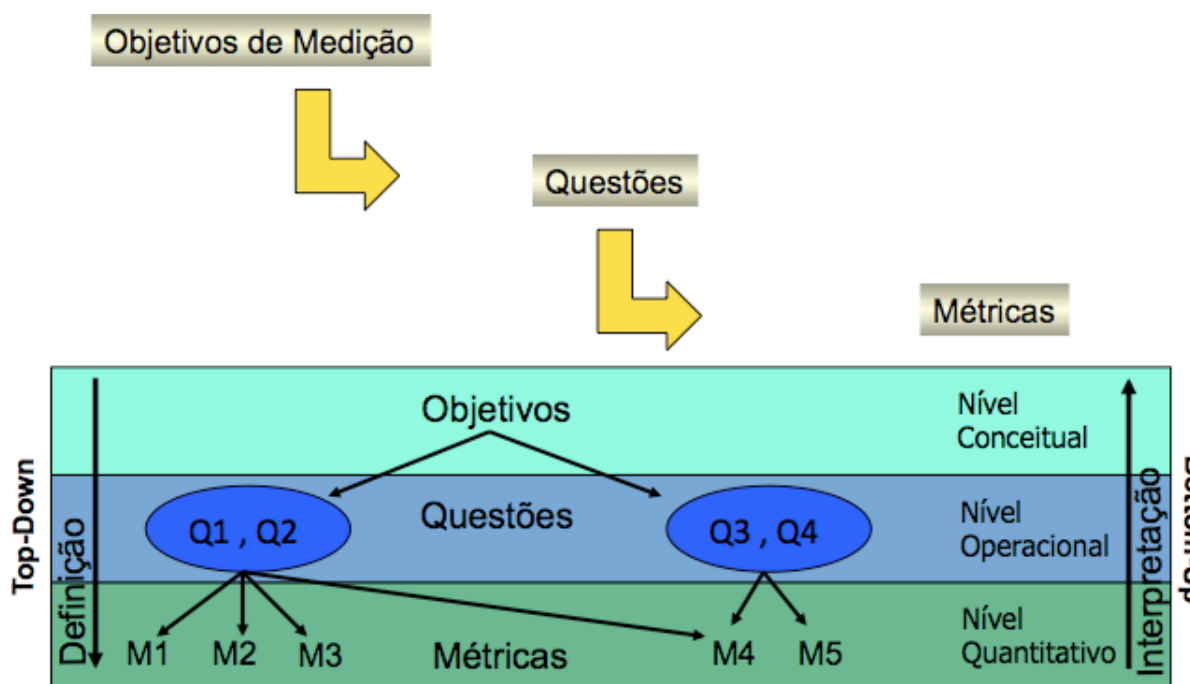


Figura 18 – Níveis do GQM (SOLLINGEN; BERGHOUT, 1999)

A Figura 18 apresenta três níveis do GQM os quais são detalhados a seguir.

1. **Nível conceitual (Objetivo):** é um objeto que diz respeito a diversos modelos de qualidade, a partir de vários pontos de vista em um dado ambiente particular. Os objetos de medição podem ser:
 - **Produtos:** são artefatos, resultados e outros documentos que são produzidos durante o processo de desenvolvimento do software. Exemplos: Especificações, casos de teste.
 - **Processos:** quaisquer atividades de software. Exemplos: Projetar, testar.
 - **Recursos:** itens que produzem resultados quando usados por processos. Exemplos: Pessoal, hardware.
2. **Nível operacional (Questionamento):** é um conjunto de perguntas que busca caracterizar se a avaliação/realização de um objetivo específico de acordo com algum modelo característico.

3. **Nível quantitativo (Métrica):** é um conjunto de dados que está associado às perguntas feitas no nível operacional. Pode ser objetiva, ou subjetiva.

- **Objetiva:** depende apenas do objeto que está sendo medido e não de opiniões particulares. Exemplo: Tamanho de um software.
- **Subjetiva:** não depende apenas do objeto que está sendo medido, mas depende também de um conjunto de opiniões particulares. Exemplo: satisfação do usuário.

A estrutura do GQM é mostrada na Figura 19:

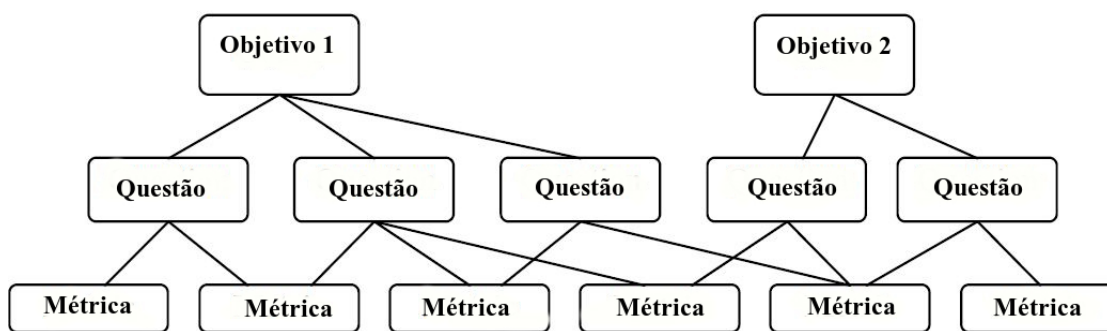


Figura 19 – Estrutura do GQM (BASILI; CALDIERA; ROMBACH, 1994)

Com esta estrutura pode-se dizer que a definição das medições se dá a partir de uma abordagem *topdown*, a partir dos objetivos de medição, questões e suas métricas correspondente, enquanto a avaliação ou análise dos resultados se dá por uma abordagem *bottom up*, a partir dos resultados das medições que respondem as questões postuladas e, por sua vez, corroboram os objetivos de medição identificados.

5.4 PSM

O *Practical Software Measurement* (PSM) é um modelo que estrutura informações com a finalidade de definir medidas que possam ser usadas em um projeto de software (MCGARRY et al., 2002 apud SANTOS et al., 2014). Patrocinado pelo departamento de defesa e pelo Exército Norte-Americano, ele tem como objetivo prover um conjunto de práticas, ferramentas e serviços para ajudar os gerentes de projetos a obter informações precisas sobre projetos que estão em andamento, para que estes atinjam suas metas de tempo, custo e qualidade (BORGES, 2003 apud MELO, S/D).

O PSM foi projetado a partir de três conceitos chaves, são eles (CARD; JONES, 2003):

- Necessidade de Informação.

- Modelo de Informação.
- Modelo de Processo de Medição.

As **necessidades de informação** são um conjunto de objetivos que servem como insumo para que os gerentes possam monitorar um determinado projeto ou processo de software. As necessidades de informação podem ser obtidas de duas formas (CARD; JONES, 2003):

1. Objetivos que o gestor deseja alcançar;
2. Obstáculos que impedem que esses objetivos sejam concretizados. Obstáculos ou questões incluem riscos, problemas e falta de informação em relação a um determinado objetivo.

O PSM organiza-se em sete necessidades de informação (CARD; JONES, 2003), conforme mostra a Tabela 9.

Tabela 9 – Detalhamento das necessidades da informação do PSM.(BAILEY et al., 2003)

Necessidade da informação	Descrição	Exemplo de Medida
Cronograma e Progresso	Relacionados ao cumprimento de marcos de projeto e à conclusão de unidades de trabalho nos prazos previstos.	Datas dos Marcos, Tempo de Folga, Unidades Codificadas.
Recursos e Custo	Relacionados à adequação entre o trabalho a ser executado e os recursos alocados ao projeto.	Orçamento, custo, Pessoal Alocado, Tempo disponível.
Tamanho e Estabilidade do Produto	Categoriza informações relacionadas à estabilidade das funcionalidades ou à capacidade requerida do software, como também ao volume necessário de software para atender a essa capacidade.	Linhas de Código, Pontos de Função.
Qualidade do Produto	Relacionada à capacidade do software produzido de atender sem falhas às necessidades do usuário.	Defeitos, Complexidade Ciclométrica, Tempo de Resposta, Conformidade com Padrões, Erros de Operação, Tempo Médio até a Falha.

Performance do Processo	Relacionada à capacidade do processo de atender às necessidades apresentadas por cada projeto.	Produtividade, Defeitos Contidos
Eficácia da Tecnologia	Trata da viabilidade e adequação das alternativas técnicas propostas, incluindo reuso, maturidade e qualidade dos componentes.	Cobertura dos Requisitos.
Satisfação do Cliente	Relaciona-se ao grau em que os produtos e serviços ofertados atendem às expectativas dos clientes.	Grau de Satisfação, Tempo de Suporte.

Cada necessidade de informação pode ser decomposta em um conceito mensurável. Um conceito mensurável é uma entidade de uma determinada necessidade de informação (CARD; JONES, 2003). A Tabela 10 mostra a necessidade de informação e seus respectivos conceitos mensuráveis.

Tabela 10 – Conceitos Mensuráveis do PSM.

Necessidade da informação	Conceito Mensurável
Cronograma e Progresso	Alcance dos Marcos. Progresso das Unidades de Trabalho. Capacidade Incremental.
Recursos e Custo	Esforço do Pessoal. Desempenho Financeiro. Ambiente e Recursos de suporte.
Tamanho e Estabilidade do Produto	Tamanho e Estabilidade Físicos. Tamanho e Estabilidade Funcionais.
Qualidade do Produto	Correção Funcional. Suportabilidade - Manutenibilidade. Eficiência. Portabilidade. Usabilidade. Dependabilidade - Confiabilidade.

Performance do Processo	Conformidade do Processo. Eficiência do Processo. Eficácia do Processo.
Eficácia da Tecnologia	Adequação da Tecnologia. Volatilidade da Tecnologia.
Satisfação do Cliente	Feedback do Cliente. Suporte ao Cliente.

Para [MCGARRY et al. \(2002\)](#), as necessidades de informação são responsáveis por definirem indicadores. Um indicador é constituído por medidas que podem ser uma ou várias medidas básicas que por sua vez servem para construir medidas derivadas. Indicadores, medidas derivadas, medidas base e entidades são usados para construir medições como mostra a Figura 20:

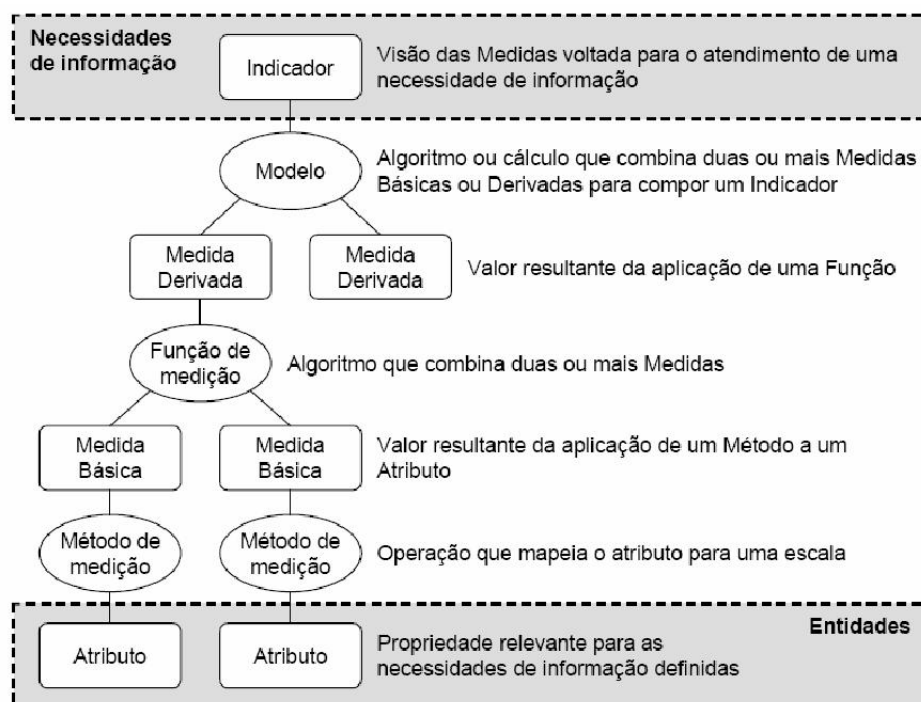


Figura 20 – Construtor de Métricas ([MCGARRY et al., 2002](#)),([BORGES, 2003](#))

O **modelo de informação** define uma relação entre as necessidades de informação mapeadas pelo gestor e os dados dos objetivos que serão coletados (medidas). O modelo de informação também fornece ideias básicas para a medição. Este modelo é definido em três níveis:

1. Medidas básicas
2. Medidas derivadas
3. Indicadores

A organização do modelo de informação do PSM é mostrada na Figura 21.

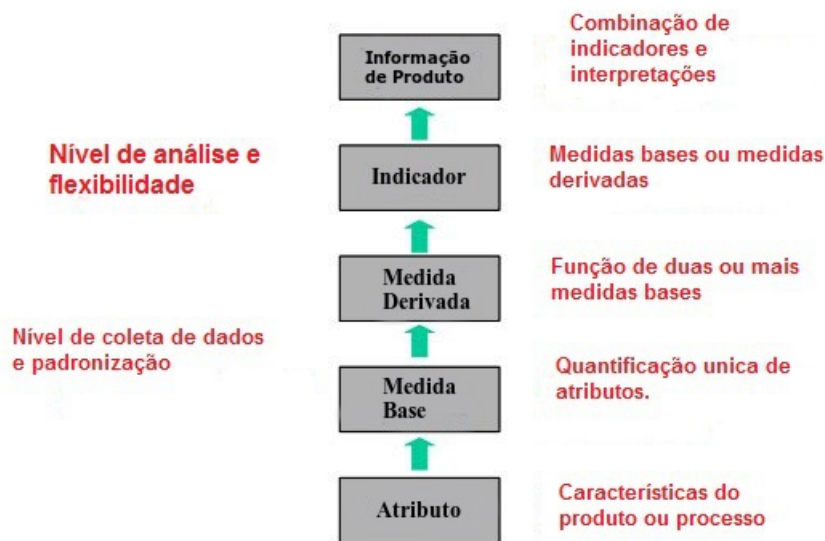


Figura 21 – Modelo de Informação do PSM (CARD; JONES, 2003)

Da perspectiva do **modelo de processo de medição**, o PSM define quatro atividades básicas, são elas (CARD, 2003):

- **Planejar Medição**: Essa atividade consiste em compreender as necessidades da informação do projeto e a definição de métricas que sejam adequadas para atender essas necessidades.
- **Executar Medição**: Essa atividade consiste na execução do modelo de informação definido na fase anterior, para que se possa chegar aos resultados esperados e que assim possa servir de insumo para que a melhor decisão possa ser tomada pela equipe de Software.
- **Avaliar Medição**: Essa atividade tem como objetivo investigar se o processo está dentro do aceitável. Para que isso seja possível é avaliar o processo de medição e avaliar informações do produto.
- **Consolidar Medição**: Essa atividade tem como objetivo obter o comprometimento da empresa em relação as medições definidas.

Desta forma o PSM é um guia bastante recomendado para empresas que desejam implantar a qualidade de software e ao mesmo tempo manter a gerência dos seus produtos.

Para o presente trabalho será usada uma abordagem GQM para a definição de métricas. A abordagem será *top-down*, e terá como objetivo determinar um conjunto de metas e medidas para gerar métricas que servirão de apoio para a seleção de produtos CMS. O GQM foi escolhido, porque é necessário neste trabalho uma profundidade grande a respeito do levantamento de métricas.

6 Aplicação

Este capítulo tem como objetivo ilustrar a concepção do método proposto para a seleção dos sistemas CMS. O raciocínio empregado para a construção do método é descrito a seguir:

1. Levantar características que estejam presentes em CMSs na literatura.
2. Relacionar as características citadas na literatura investigada para o estudo com as características SQuaRE.
3. Elaborar um *survey* para avaliar a relevância das características identificadas.
4. Submeter *survey* para especialistas.
5. Analisar resultados do *survey* e mapear características mais relevantes.
6. Estruturar a primeira parte do método baseado nas características mais relevantes que não podem ser mensuradas pela SQuaRE.
7. Estruturar a segunda parte do método baseado nas características mais relevantes que podem ser mensuradas a partir da Square.
8. Submeter um segundo *survey* para especialistas com a finalidade de validar o método proposto.
9. Avaliar resultados obtidos com o *survey*.

6.1 Identificação das características

Nesta seção é apresentado o passo a passo para a seleção das características encontradas. Nesta etapa foram considerados os artigos identificados a partir das expressões de busca detalhadas no Apêndice A e alguns artigos usados no Capítulo 3 para o referencial teórico.

6.1.1 Características de CMSs para os mais populares

Os artigos identificados apresentam características de CMSs populares Joomla, Wordpress, Drupal. A Tabela 21 do Apêndice B mostra os artigos relacionados aos CMSs mais populares. A partir destes artigos, foram obtidas as características que serão usadas para a elaboração do método. Deve ser observada nessa tabela a coluna “id”, que representa um código identificador do artigo levantado. O código do artigo será usado na Seção

6.1.3 (Rastreabilidade de características), a fim de estabelecer uma rastreabilidade entre artigos e características levantadas.

Desta forma, a Tabela 21 apresenta os contextos de artigos para os CMSs populares. A partir desta tabela foi construída a Tabela 23 do Apêndice B, com as características de CMS populares. Estas tabelas apoiaram a elaboração das tabelas apresentadas nas Figuras 54 e 55 que, por sua vez serão explicadas na sequência deste trabalho.

6.1.2 Características de CMSs para os não populares

Para uma análise mais completa de características que refletisse a maior variedade de CMSs possíveis foi feito um mapeamento de artigos para os CMSs não populares. A Seção A.2 do apêndice A apresenta como foi feito este mapeamento. Além disso, nesta etapa foi considerado apenas CMSs *Open Source* ou Software Livre.

A partir dos resultados fornecidos por meio da expressão de busca para CMSs não populares foram elaboradas as tabelas 22 e 24, que representam respectivamente os contextos para os artigos levantadas e as novas características encontradas.

Desta forma, as tabelas 22 e 24 apoiaram a elaboração das tabelas apresentadas nas figuras 56, 57 que, por sua vez, serão explicadas na sequência deste trabalho.

6.1.3 Rastreabilidade de características

A medida que os artigos da tabela 21 e da tabela 22 eram lidos e as características apresentadas nas tabelas 23 e 24 eram encontradas foi feito a rastreabilidade de quais características eram citadas nos artigos. As imagens 54, 55, 56, 57 mostram essa rastreabilidade.

Para interpretar as imagens 54, 55, 56, 57 relacione os números apresentados na linha horizontal com o id apresentado nas tabelas 21 e 22, ou seja, a linha horizontal de números diz respeito aos artigos apresentados. Os números apresentados na coluna vertical devem ser relacionados com os id apresentados nas tabelas 23 e 24, ou seja a coluna vertical de números diz respeito as características de CMSs encontradas.

Exemplo:

- Na figura 54 a linha 10 está mapeada com as colunas 3, 7 e 9. Isso quer dizer que a característica 10 - liberdade para alteração de um template específico é citada nos artigos 3, 7, 9 ... , que são os artigos Reis et al. (2011) , Xiang e Yu (2010b) e Cheng e Wang (2012).
- De forma semelhante na figura 57, o artigo número 8 se relaciona com as linhas 47, 48, 51 da tabela 23 e as linhas 7, 9, 10, 11 se relacionam com a tabela 24, isto é o

artigo [Lina e Guohua \(2010\)](#) cita todas estas características.

A rastreabilidade para todos os artigos lidos é mostrado nas figuras [54](#), [55](#), [56](#), [55](#) presentes na Seção [B.5](#) do Apêndice [B](#).

Após a construção das figuras [54](#), [55](#), [56](#), [55](#) foram observadas as seguintes conclusões.

- Em 43 artigos lidos referentes aos CMSs mais populares (Joomla, Wordpress, Drupal) foram identificadas 57 características. Além disso, foram identificadas mais 15 características em 12 artigos para os CMSs não populares resultando em um total de 72 características identificadas.
- Para os CMSs mais populares, o artigo em que mais características foram citadas foi o Artigo de Id numero "24" de [Serodio et al. \(2010\)](#), com 21 citações. Logo depois, o maior número de citações foi do artigo de Id número "11" de [Marimuthu e Sekaran \(2013\)](#), com 19 citações.
- Para os CMSs não populares foi observado que o artigo que mais contribuiu com citações de características foi o artigo de Id número "3" de [Loechel, Mihelcic e Pickl \(2012\)](#), com 29 características. Logo depois o artigo que mais contribuiu foi o de [Nath e Arora \(2010\)](#), com 26 citações.

6.1.4 Contagem de características e mapeamento com as características SQuaRE

A Tabela [25](#) no Apêndice [B](#) mostra um cruzamento das características de CMSs levantadas, com características da norma [ISO/IEC-25000 \(2005\)](#). Esta tabela reflete o contexto de qualidade do produto. Além disso, é apresentada na coluna "Qtd" a quantidade de citações de cada característica somados os artigos apresentados na Tabela [21](#) e na Tabela [22](#).

A partir do mapeamento feito nesta seção foram estabelecidas as seguintes conclusões.

- As cinco características mais citadas foram **licenciamento** (40 citações), **facilidade para interação com várias ferramentas** (38 citações), **a Modularidade** (32 citações), **uso de bancos de dados relacionais** (28 citações) e **existência de recursos para controle de acesso** (25 citações).
- Características de qualidade do produto e de qualidade em uso apareceram. As características de qualidade do produto foram mais frequentes.

- Do que diz respeito a qualidade do produto, as características da SQuaRE que apareceram foram **Adequação Funcional**, **Usabilidade**, **Portabilidade**, **Manutenibilidade**, **Segurança** e **Eficiência de Desempenho**. A característica mais frequente foi a **Adequação Funcional**.
- Do que diz respeito a qualidade em uso, as características SQuaRE que apareceram foram a **Cobertura de Contexto** e a **Inexistência de Riscos**.
- Foram identificadas características que não podem ser mensuradas pela SQuaRE com número de citações relevantes. Exemplo: "Ser software livre ou não"(40 citações).

6.2 Aplicação do Primeiro Questionário

Com o objetivo de saber quais das características mapeadas na seção 6.1 são relevantes para o estudo foi feito um questionário. O questionário foi submetido a especialistas e desenvolvedores que trabalham com CMS em comunidades espalhadas na rede social Facebook. Este questionário é descrito no Apêndice C e detalhado quanto a aplicação na Seção 2.5.2 (Instrumentos para a coleta de dados) deste trabalho.

Para a construção das perguntas foi considerado o mapeamento de características construído na Tabela 25 no Apêndice B. As características foram agrupadas quanto ao número de citações e a partir dos resultados foram observadas quais características apareciam com mais frequência. Para cada característica que aparecia com frequência foi feita uma pergunta mista, com itens que eram compostos pelas características de CMSs mais citadas na Tabela 25 do Apêndice B.

6.2.1 Critérios para análise de resultados do primeiro questionário

Para a análise de dados do Questionário 1 do Apêndice C foram considerados os seguintes critérios:

1. Perguntas de múltipla escolha com mais de 12 itens: foram classificados os itens com mais de 75 % de aderência do total. Exemplo: Questão 3.
2. Perguntas de múltipla escolha com quantidade de itens entre 8 e 12: foram classificados os itens com aderência de 60 %.
3. Perguntas de múltipla escolha com menos de 8 itens: foram classificados os itens com mais de 50 % de aderência. Exemplo: Questão 7.
4. Para a questão 13 devido a complexidade das características envolvidas foi adotado $> = 65$ % de aderência.

6.2.2 Análise dos resultados das questões do questionário

Nesta seção serão apresentados os resultados para o Questionário 1 aplicado com o objetivo de filtrar as características mais relevantes.

1) No momento de escolher um determinado CMS para uma aplicação você leva em conta o Licenciamento?

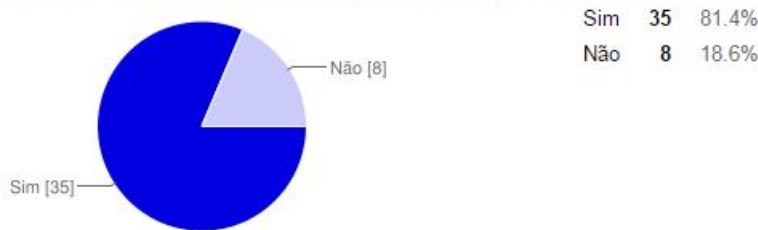


Figura 22 – Resultado da Questão 1 para o Questionário 1

A figura 22 mostra os resultados para questão 1 do questionário que pergunta sobre o licenciamento de software. Nesta questão 81,4% dos 43 participantes dizem se preocupar com o licenciamento na hora de escolher um determinado CMS, por ter tido uma aceitação muito grande (superior a 75 %) essa foi uma característica considerada importante.

2) Você prefere um CMS software Livre ou software proprietário para desenvolver suas aplicações?

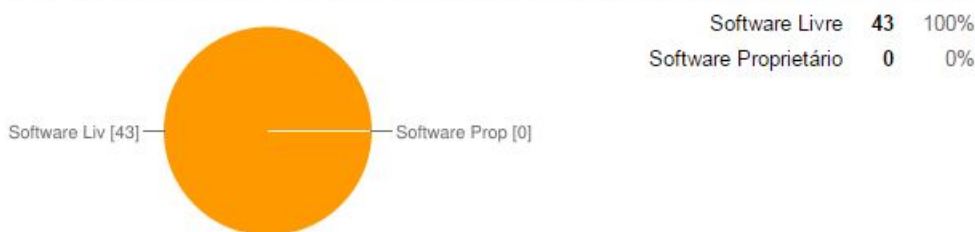


Figura 23 – Resultado da Questão 2 para o Questionário 1

O resultado para a Questão 2 expresso na figura 23 foi enviesado, pelo fato de que o questionário foi aplicado em comunidades de CMS software livre, e uma comunidade de desenvolvimento web, logo já se esperava que o resultado fosse 100 % software livre. Se o questionário tivesse sido aplicado em uma determinada empresa que usa CMS proprietário o resultado seria diferente. Porém os 43 entrevistados justificaram a escolha dos CMSs software livre com algumas justificativas levantadas na subquestão 2.1. Essas justificativas em resumo foram:

- Custos com licença
- Apoio da comunidade que ajuda de forma colaborativa com o crescimento do CMS
- Liberdade para customizar o CMS

- Experiência

Um *feedback* de um dos entrevistados foi "*Criar um bom CMS do zero leva tempo e custa muito dinheiro, sendo que existem soluções maduras, prontas para usar, e altamente testadas no mundo do Software Livre*" que justifica os itens a cima.

3) No que diz respeito às características funcionais do CMS, quais delas você julga como importantes ou indispensáveis em um CMS?

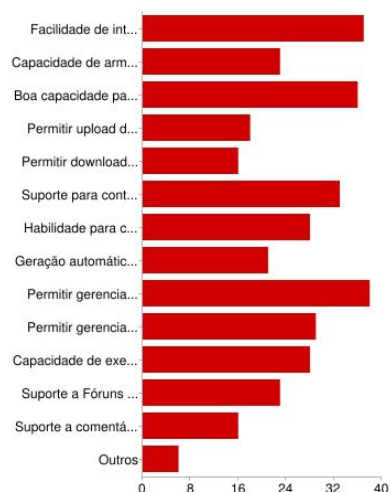


Figura 24 – Resultado da Questão 3 para o Questionário 1

Facilidade de interação do CMS com várias ferramentas. (ex: outras linguagens, bancos de dados, frameworks de programação, etc)	37	86%
Capacidade de armazenar dados em bancos de dados relacionais.	23	53.5%
Boa capacidade para realizar pesquisas ou buscas de conteúdos no próprio site.	36	83.7%
Permitir upload de uma grande quantidade de tipos de arquivos Exemplos: (Word, Excel, RTF, PDF, etc).	18	41.9%
Permitir download de uma grande quantidade de tipos de arquivos Exemplos: (Word, Excel, RTF, PDF, etc).	16	37.2%
Suporte para conteúdos multimídia (ex: vídeos, músicas, etc).	33	76.7%
Habilidade para controlar e gerenciar múltiplas versões do mesmo conteúdo.	28	65.1%
Geração automática de Interface de Usuário.	21	48.8%
Permitir gerenciamento de usuários.	38	88.4%
Permitir gerenciamento de arquivos.	29	67.4%
Capacidade de executar Backup dos conteúdos do site.	28	65.1%
Suporte a Fóruns de discussão.	23	53.5%
Suporte a comentários.	16	37.2%
Outros	6	14%

Figura 25 – Resultado da Questão 3 por percentual para o Questionário 1.

As figuras 24 e 25 estão relacionadas a característica de Adequação Funcional da SQuaRE. Para esta pergunta foram selecionadas 15 características, a fim de perguntar ao entrevistado qual ou quais características são mais importantes em um CMS. Destacaram-se como melhores resultados para essa pergunta:

- Facilidade que o CMS possui para interagir com várias ferramentas
- Capacidade para realizar buscas no próprio site
- Suporte para conteúdos multimídias

- Permitir o gerenciamento de usuários

Para a Questão 3, foram considerados resultados satisfatórios, as características que obtiveram mais de 75 % de aderência. Além desses resultados no campo outros destacou-se as seguintes respostas:

- Suporte para e-commerce
- Extensibilidade

4) A quantidade de templates ou plugins disponíveis para um determinado CMS é um fator importante para escolha desse CMS?

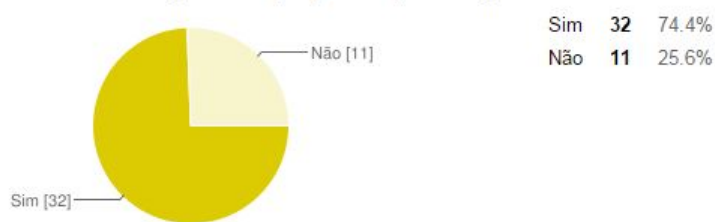


Figura 26 – Resultado da Questão 4 para o Questionário 1

Para a Questão 4, conforme mostra a Figura 26, na opinião dos entrevistados, a quantidade de *templates* e *plugins* é importante para escolha de produtos CMS.

5) No que diz respeito a Usabilidade, quais características você julga como importantes ou indispensáveis na hora de escolher um CMS?

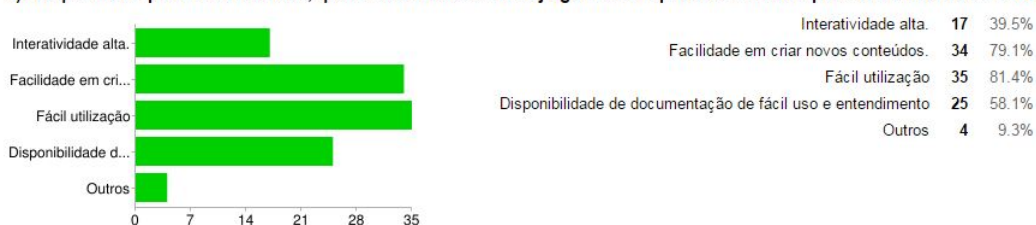


Figura 27 – Resultado da Questão 5 para o Questionário 1

Na Questão 5, conforme mostra a Figura 27 foi perguntado aos entrevistados sobre a característica de Usabilidade. Nesse quesito destacou-se:

- Fácil Utilização
- Facilidade em criar novos conteúdos
- Disponibilidade de documentação de fácil uso e entendimento

Nesta pergunta, foram considerados os itens que obtiveram mais de 50 % de aceitação, além disso no campo outros. Os entrevistados sugeriram:

- Editor *wysiwyg*¹ robusto.
- Padronização da Interface e do Código
- Curva de aprendizado

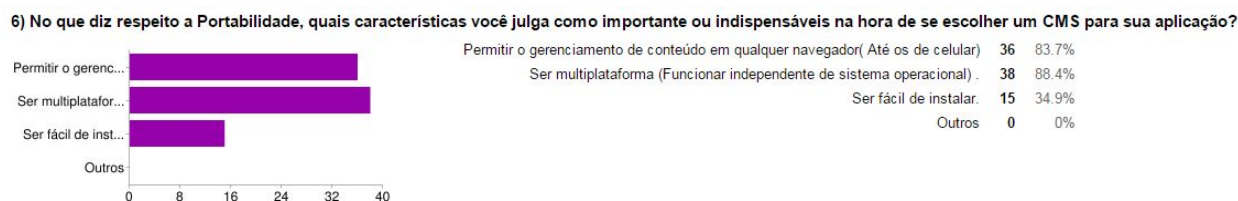


Figura 28 – Resultado da Questão 6 para o Questionário 1

A Figura 28 mostra os resultados para a característica de Portabilidade. Dois resultados se destacaram com mais de 50 % foram eles:

- Permitir o gerenciamento de conteúdo de qualquer navegador
- Ser multiplataforma

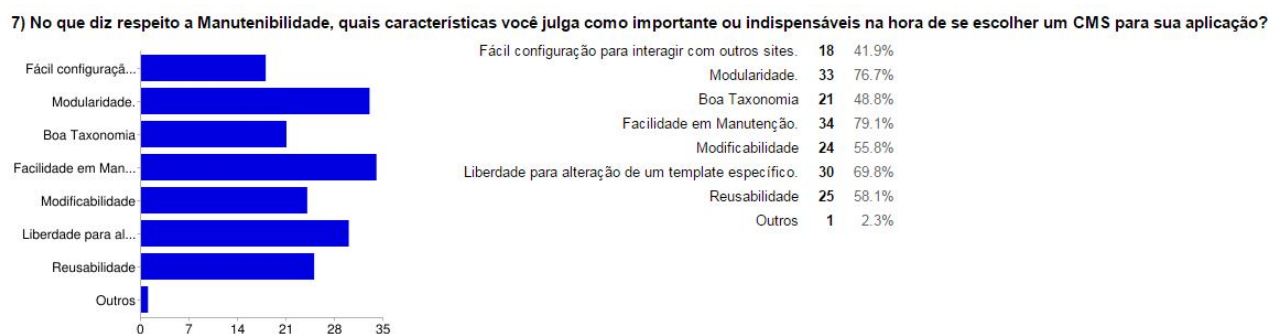


Figura 29 – Resultado da Questão 7 para o Questionário 1

A Questão 7 aborda a característica de Manutenibilidade. De acordo com a Figura 29, cinco itens se destacaram com mais de 50 % de aderência. Esses itens foram:

- Modularidade
- Facilidade em Manutenção
- Modificabilidade
- Liberdade para alteração de um template específico
- Reusabilidade

¹wysiwyg = What you see is what you get.

Um entrevistado disse no campo "outros" que é importante um CMS possuir um *Framework* de testes.

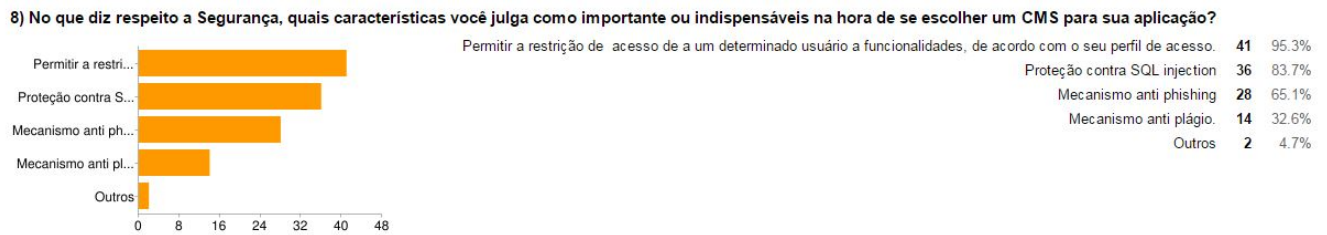


Figura 30 – Resultado da Questão 8 para o Questionário 1

A Figura 30 aborda a característica de Segurança da SQuaRE. Três itens se destacaram com mais de 50 %. Foram eles:

- Permitir a restrição de acesso de um determinado usuário
- Proteção contra *SQL Injection*
- Mecanismo anti *phishing*

Também foi dito no campo "outros" a característica: "Segurança do código fonte".

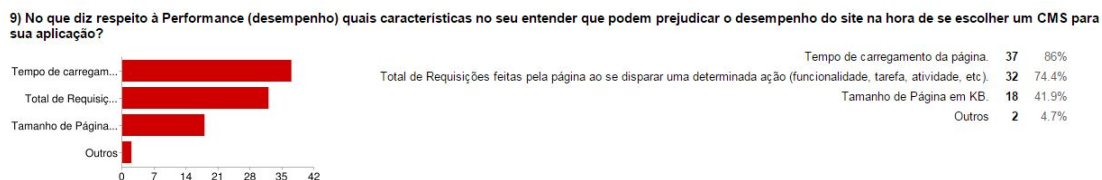


Figura 31 – Resultado da Questão 9 para o Questionário 1

De acordo com a Figura 31, no que diz respeito ao Desempenho, duas características se destacaram com mais de 50 % de aderência. Essas características foram:

- Tempo de carregamento da página
- Total de requisições

No campo outros foi sugerido:

- Quantidade de relacionamentos entre os conteúdos
- Configuração de Cache;

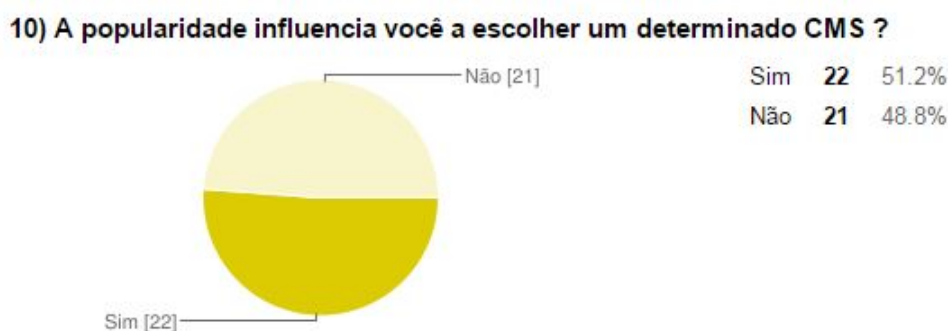


Figura 32 – Resultado da Questão 10 para o Questionário 1

A Figura 32 mostra o resultado para a característica de popularidade. Dos 43 entrevistados, 22 disseram que "Sim" a popularidade influencia na escolha de um determinado CMS; enquanto 21 entrevistados disseram que não influenciam. É possível ver portanto certo equilíbrio nas opiniões dos entrevistados. As justificativas dos entrevistados do porquê a popularidade influencia na escolha de um CMS foram:

- Maior agilidade para resolver questões e aprimorar o CMS
- Quanto mais popular, maior a comunidade e melhor a troca de experiências e o suporte oferecido pelos desenvolvedores
- Maior atração para pessoas ajudarem a contribuir com a evolução do CMS
- Quanto mais popular, mais recursos são desenvolvidos
- Apesar de popularidade não implicar em qualidade, dependendo do projeto na maioria das vezes a seleção dos CMS's candidatos acontece pela popularidade
- Um CMS popular traz mais confiança em relação ao seu uso
- CMSs mais populares terão mais plugins disponíveis. Isso não torna o CMS melhor, ou garante a qualidade dos plugins, mas é um grande fator para se escolher qual CMS utilizar. Além disso, a comunidade pode ser maior, facilitando possíveis pedidos de suporte

Já os entrevistados que opinaram de forma negativa quanto a popularidade justificaram com os argumentos a seguir:

- Desde que o CMS disponha de uma boa documentação e cumpra aquilo a que se propõe a popularidade não será ou (deverá) ser um fator importante
- Um sistema mais popular chama mais a atenção de hackers

- A adequação do CMS ao foco do projeto é mais importante do que a popularidade da ferramenta escolhida
- Alguns CMS populares não oferecem recursos avançados
- Popular não significa que se tenha boa qualidade
- A usabilidade é um fator mais relevante que a popularidade

11) A arquitetura de um CMS é um fator preponderante para a sua escolha ?

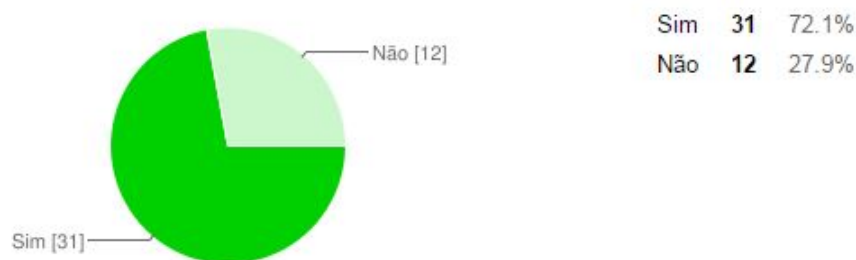


Figura 33 – Resultado da Questão 11 para o Questionário 1

A Figura 33 mostra os resultados da Questão 11 a respeito da arquitetura dos CMSs. 72,1 % dos entrevistados disseram que sim a arquitetura é um fator que pesa na hora de se escolher um determinado CMS. Em resumo as justificativas dos entrevistados foram as apresentadas a seguir:

- Uma boa arquitetura ajuda no entendimento de como o CMS funciona
- Facilita o uso e aplicação do CMS em questão
- Define se o site feito com CMS em questão, será extensível ou não
- Facilita a manutenção
- Uma arquitetura extensível e de fácil manutenção é essencial para o desempenho de um CMS robusto. Uma má arquitetura pode inviabilizar o uso do CMS no caso de sites com uma grande quantidade de dados e/ou grande quantidade de page views

12) O tamanho da comunidade de pessoas que usam o CMS é um fator preponderante para a escolha de um determinado CMS?

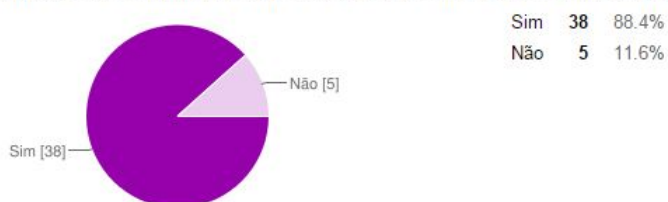


Figura 34 – Resultado da Questão 12 para o Questionário 1

A Figura 34 mostra o resultado para a Questão 12 que diz respeito ao tamanho da comunidade de um determinado CMS. 82 % dos entrevistados disseram que "Sim" o tamanho da comunidade pesa ao se escolher um determinado CMS. As justificativas para o resultado são apresentadas a seguir:

- Melhor suporte
- Muitas pessoas trabalham em prol de objetivos comuns
- Troca de experiências
- Melhor índice de confiabilidade, devido à muitos usuários utilizando a mesma plataforma.
- Continuidade do projeto

13) Qua(l)is das características abaixo você julga também importante ter em um CMS, além das já perguntadas?

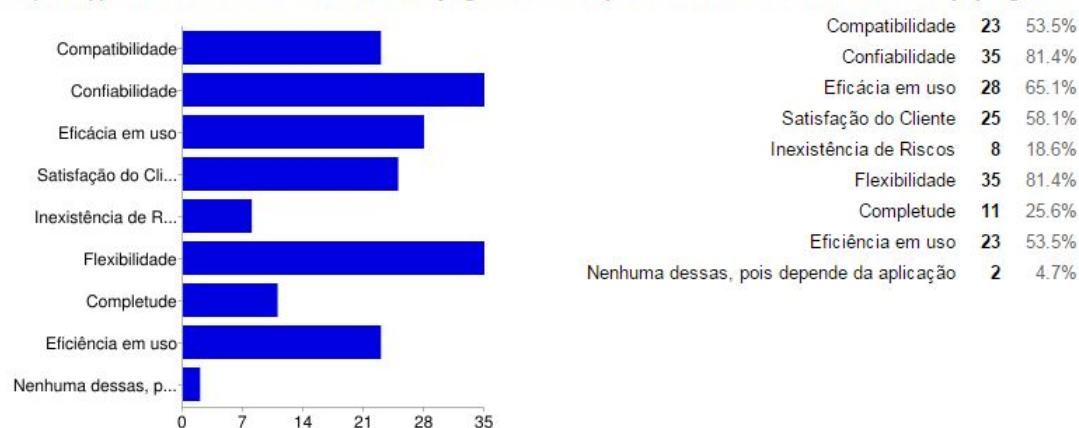


Figura 35 – Resultado da Questão 13 para o Questionário 1

A Figura 35 mostra os resultados para a Questão 13, que pergunta aos entrevistados sobre as características da SQuaRE que não foram mapeadas na literatura. A subcaracterística de Flexibilidade foi inserida nesta questão, pois obteve um grande número de citações na literatura. Foram considerados os resultados com mais de 65 % de resposta. Esses resultados são apresentados a seguir:

- Flexibilidade
- Confiabilidade
- Eficácia em Uso

Por fim a Questão 14 perguntou aos usuários/especialistas em CMS, se existem mais características importantes e que devem ser levadas em consideração na hora de se escolher um determinado CMS. Em resumo as respostas apresentadas foram:

- Integração de dados via XML
- Layouts leves e responsivos
- Permitir construção de aplicações *mobiles*
- Extensibilidade

6.3 Visão Geral do Método

O método será composto por duas partes. A primeira parte diz respeito às características que não podem ser mensuradas pela SQuaRE. Já a segunda parte diz respeito às características que podem ser medidas pela SQuaRE. Em ambos os casos serão usadas as características aprovadas e as características sugeridas pelos especialistas na seção anterior.

A Figura 36 mostra um fluxograma de execução do método proposto. Os passos vistos na figura serão explicados nas Seções 6.4.2 e 6.5.

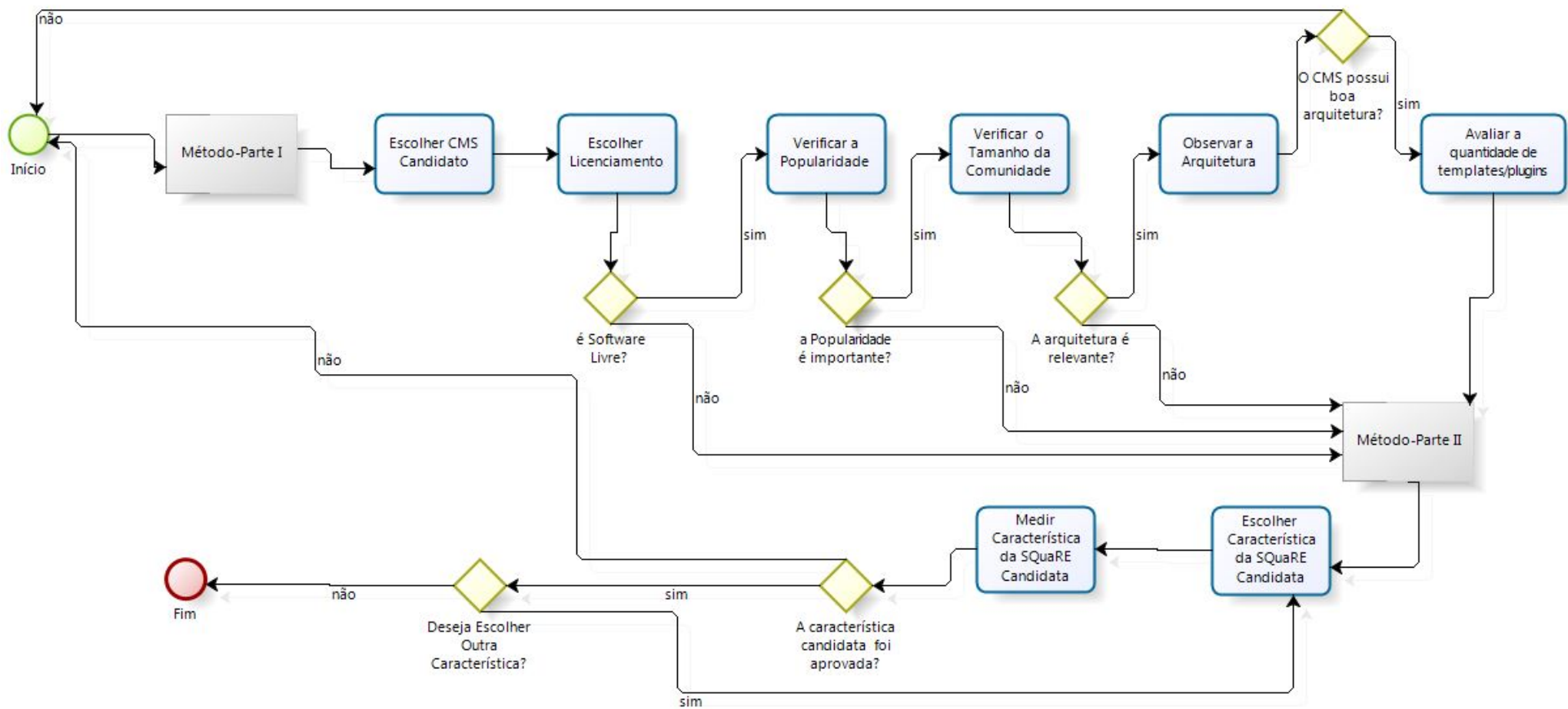


Figura 36 – Fluxograma do método proposto

6.4 Construção do Método - Parte I

Esta seção tem como objetivo organizar a primeira parte do método que será composta por características que não podem ser mapeadas na SQuaRE.

6.4.1 Listagem de características aprovadas pelos especialistas

Após a aplicação do Questionário 1 (Levantamento de Características de CMS) para os especialistas e a discussão dos resultados, as características que não puderam ser mapeadas pela SQuaRE, mas que foram consideradas importantes foram:

1. Licenciamento.
2. Popularidade.
3. Tamanho da Comunidade.
4. Arquitetura do CMS.
5. Quantidade de Templates e Plugins disponíveis.

Estas características foram selecionadas para comporem o método na parte 1.

6.4.2 Passo a passo para a escolha do CMS - Parte I

Esta seção explica como as características listadas no item anterior, relacionam-se para formar uma sequência de passos para o auxílio da escolha de um determinado CMS. O passo a passo apresentado a seguir e corresponde aos passos apresentados na Figura 36:

Passo 1: Escolher CMS candidato. A Seção 3.4.1 apresenta vários CMSs que podem ser escolhidos, mas existem outras opções que podem também ser usadas, mas que não constam nesta seção.

Passo 2: Verificar qual o Licenciamento do CMS em questão.

No que diz respeito ao Licenciamento está sendo considerado apenas o fato do CMS ser Software Livre ou Software Proprietário. A Seção 3.4.1 deste trabalho explica as principais diferenças entre os dois tipos de software e apresenta alguns exemplos de CMSs software livre e CMSs software proprietário.

De acordo com os resultados obtidos na Seção 6.2.2 do *survey* com os especialistas, a escolha de um CMS software livre é uma boa escolha pelos seguintes motivos:

- Custos com licença

- Apoio da comunidade que ajuda de forma colaborativa com o crescimento do CMS
- Liberdade para customizar o CMS
- Experiência

Porém, caso as preocupações com o custo de licenças sejam irrelevantes, ou não se tenha interesse em nenhum dos fatores citados anteriormente, deve-se prosseguir diretamente para o primeiro passo da segunda parte do método na seção 6.5.

Passo 3: Verificar a popularidade do CMS em questão.

Baseado na escolha de um CMS Software Livre neste passo deve-se analisar a popularidade do CMS em questão.

A Seção 3.4.2 deste trabalho faz uma apresentação dos CMSs mais populares e do porquê a popularidade pode ser importante para um CMS Software Livre. Para os especialistas, a popularidade de um determinado CMS se faz importante pelos motivos apresentados na seção 6.2.2. Porém, conforme foi apontado pelos especialistas na mesma seção, a popularidade pode também não ser um critério relevante na escolha de um determinado CMS. Alguns dos motivos levantados pelos especialistas para que a popularidade não seja levada em consideração:

- Desde que o CMS disponha de uma boa documentação e cumpra aquilo a que se propõe a popularidade não será ou (deverá) ser um fator importante.
- Um sistema mais popular chama mais à atenção de hackers.
- A adequação do CMS ao foco do projeto é mais importante do que a popularidade da ferramenta escolhida.
- A usabilidade é mais relevante que a popularidade.

Para se escolher um CMS com base na popularidade podem ser usados um dos três exemplos fornecidos na seção 3.5.

Caso a popularidade seja considerada importante, deve-se prosseguir para o Passo 4 desta seção, caso contrário siga para o passo 5.

Passo 4: Verificar o Tamanho da Comunidade do CMS em Questão.

A Tabela 2 e as Figuras 4, 5, 6, 7, 8 e 9 mostram alguns números para tamanho de comunidade de CMSs levando em consideração redes sociais e o número de *commits* do repositório das aplicações. Além disso, conforme mostra a Figura 34, mais de 80 % dos especialistas entrevistados citaram como fatores importantes:

- Oferecer um melhor suporte
- Troca de Experiências
- Melhor índice de confiabilidade, devido à muitos usuários utilizando a mesma plataforma

Porém caso o tamanho da comunidade não seja importante, deve-se desconsiderar este passo e prosseguir para o passo 5 desta seção.

Passo 5: Observar a arquitetura do CMS em questão.

Neste passo existem dois pontos de decisão chave. O primeiro pergunta ao desenvolvedor se a arquitetura é um fator relevante para a aplicação. Caso seja importante o desenvolvedor deverá investigar a arquitetura do CMS escolhido.

Na Questão 33 os especialistas julgaram que a arquitetura de um CMS se faz importante pelos motivos apresentados na Seção 6.2.2, estes motivos foram:

- Facilitar o uso e aplicação do CMS em questão
- Definir se o site feito com o CMS em questão, será extensível ou não
- Facilitar a manutenção

O segundo ponto chave pergunta se o CMS possui boa arquitetura. Caso tenha o desenvolvedor poderá ir para o próximo passo. Se o CMS não possuir boa arquitetura o desenvolvedor volta para o início do fluxograma, pois uma vez considerada como característica importante pelo desenvolvedor, não sendo aprovada é sinal de que o CMS não será bom para aquele determinado contexto.

Não foram sugeridas métricas para este quesito. Cabe ao desenvolvedor escolher como mensurar e concluir se a arquitetura do CMS em questão é adequada para a sua aplicação.

Passo 6: Avaliar a quantidade de templates e plugins disponíveis no CMS.

Caso este seja um fator relevante para o sucesso da aplicação em questão o próprio responsável por desenvolver a aplicação deve considerar se a quantidade de templates e plugins é relevante para o CMS escolhido.

6.5 Construção do Método - Parte II

Nesta parte do método o objetivo será mapear as características que podem ser medidas com o auxílio das normas SQuaRE. Para esta etapa foi realizado um GQM para identificar e sugerir métricas para as características estabelecidas como relevantes.

6.5.1 Listagem de características aprovadas pelos especialistas

A Tabela 11 apresenta as características relacionadas a qualidade do produto que foram aprovadas pelos especialistas de acordo com a Seção 6.2.2. As notas de rodapé que explicam as subcaracterísticas mapeadas são referenciadas de acordo com a ISO/IEC-25023 (2011).

A tabela 11 foi baseada na Tabela B.4 do Apêndice B, no qual características de CMSs foram relacionadas com suas respectivas características SQuaRE, a fim de identificar quais características necessitam serem medidas.

Tabela 11 – Cruzamento de Características aprovadas pelos especialistas com características SQuaRE - Qualidade do Produto.

Características de CMS aprovadas	Características Square	Subcaracterísticas Square
Facilidade que o CMS possui para interagir com várias ferramentas	Adequação Funcional	Completude Funcional ²
Capacidade para realizar buscas no próprio site.	Adequação Funcional	Completude Funcional
Suporte para conteúdos multímidia	Adequação Funcional	Completude Funcional
Integração de dados via XML	Adequação Funcional	Completude Funcional
Permitir construção de aplicações <i>mobiles</i>	Adequação Funcional	Completude Funcional
Permitir o gerenciamento de usuários	Adequação Funcional	Completude Funcional
Suporte para e-commerce	Adequação Funcional	Completude Funcional
Extensibilidade	Adequação Funcional	Completude Funcional
Editor wisiwig robusto	Adequação Funcional	Completude Funcional
Fácil Utilização	Usabilidade	Operacionalidade ³
Disponibilidade de documentação de fácil uso e entendimento	Usabilidade	Aprendizado ⁴

²Subcaracterística que estabelece que um conjunto de funcionalidades deve abranger todas as tarefas e objetivos específicos dos seus usuários.

³Subcaracterística que estabelece que um determinado produto ou sistema possui atributos que o tornam fácil de operar e controlar.

⁴Subcaracterística que estabelece que um produto ou sistema pode ser usado por usuários específicos para atingir metas de aprendizado ao se utilizar um produto com efetividade, eficiência, inexistência de riscos e satisfação em um determinado contexto de uso.

Facilidade em criar novos conteúdos	Usabilidade	Operacionalidade
Curva de Aprendizado	Usabilidade	Aprendizado
Layouts Leves e Responsivos	Usabilidade	Estética ⁵
Permitir o gerenciamento de conteúdo de qualquer navegador	Portabilidade	Adaptabilidade ⁶
Ser multiplataforma	Portabilidade	Adaptabilidade
Modularidade	Manutenibilidade	Modularidade ⁷
Modificabilidade	Manutenibilidade	Modificabilidade ⁸
Liberdade para alteração de um template específico	Manutenibilidade	Modificabilidade
Reusabilidade	Manutenibilidade	Reusabilidade ⁹
Padronização da interface e do código	Manutenibilidade	Reusabilidade
<i>Framework</i> de testes	Manutenibilidade	Testabilidade ¹⁰
Permitir a restrição de acesso de um determinado usuário	Segurança	Confidencialidade ¹¹
Proteção contra <i>SQL Injection</i>	Segurança	Integridade ¹²
Mecanismo anti <i>phishing</i>	Segurança	Integridade
Segurança do código fonte	Segurança	Integridade
Tempo de carregamento da página	Eficiência de Desempenho	Comportamento do Tempo ¹³

⁵Subcaracterística que estabelece que a interface de usuário deve permitir uma interação agradável e satisfatória para o usuário.

⁶Subcaracterística que estabelece que um produto ou sistema pode ser adaptado de forma eficaz e eficiente para um hardware, software, ou outros ambientes operacionais de uso.

⁷Subcaracterística que estabelece que um software, ou sistema possui componentes discretos, tais que a mudança para um componente tem um impacto mínimo sobre outros componentes.

⁸Subcaracterística que estabelece que um produto ou sistema pode ser modificado de forma eficaz e eficiente sem introduzir defeitos.

⁹Subcaracterística que estabelece que um componente pode ser utilizado em mais do que um sistema, ou na construção de outros ativos.

¹⁰ Subcaracterística que estabelece a possibilidade de determinar critérios de teste para um dado produto, sistema, ou componente.

¹¹ Subcaracterística que estabelece que um produto ou sistema garante que os dados são acessíveis somente por pessoas autorizadas ao acesso.

¹²Subcaracterística que estabelece que um sistema, produto ou componente deve impedir o acesso não autorizado ou a modificação de dados.

¹³Subcaracterística que estabelece que os tempos de resposta, de processamento e as taxas de trans-

Total de requisições	Eficiência de Desempenho	Utilização de Recursos
Quantidade de relacionamentos entre os conteúdos	Eficiência de Desempenho	Utilização de Recursos ¹⁴
Configuração de Cache	Eficiência de Desempenho	Utilização de Recursos
Confiabilidade	Confiabilidade	Maturidade ¹⁵ , Tolerância a Falhas ¹⁶ Capacidade de Recuperação ¹⁷ e Disponibilidade ¹⁸

De forma semelhante a Tabela 12 apresenta as características de qualidade em uso aprovadas pelos especialistas.

Tabela 12 – Cruzamento de Características aprovadas pelos especialistas com características SquaRE - Qualidade em uso.

Características de CMS aprovadas	Características Square	Subcaracterísticas Square
Flexibilidade	Cobertura de Contexto	Flexibilidade ¹⁹
Efetividade	Efetividade	Efetividade

6.5.2 Passo a passo para a escolha do CMS - Parte II

Passo 1: Escolher Característica(s) SQuaRE candidata(s).

O objetivo deste passo é escolher entre uma, ou várias das características listadas abaixo para serem medidas no passo 2.

De acordo com as tabelas 11 e 12 as características SQuaRE que apareceram foram:

- Adequação Funcional

ferência de um produto ou sistema devem atender aos requisitos.

¹⁴Subcaracterística que estabelece que as quantidades de recursos utilizados por um produto, ou sistema devem atender aos requisitos.

¹⁵Subcaracterística que estabelece que um sistema deve satisfazer as necessidades de confiabilidade em operação normal.

¹⁶Subcaracterística que estabelece que um sistema, produto ou componente deve operar como pretendido, apesar da presença de falhas de hardware ou software.

¹⁷Subcaracterística que estabelece que em caso de uma interrupção ou falha, um produto ou sistema pode recuperar os dados diretamente afetados e restabelecer o estado desejado do sistema.

¹⁸Subcaracterística que estabelece que um sistema, produto, ou componente está operacional e acessível quando necessário para uso.

¹⁹Subcaracterística que estabelece o quanto um produto ou sistema pode ser utilizado de forma eficaz, de forma eficiente, livre de riscos e com satisfação em contextos além daqueles definidos inicialmente na especificação de requisitos. (ISO/IEC-25022, 2012)

- Usabilidade
- Manutenibilidade
- Portabilidade
- Segurança
- Eficiência de Desempenho
- Confiabilidade
- Eficácia em Uso ou Efetividade
- Cobertura de Contexto

Neste passo, antes de escolher uma destas características, observe os seguintes questionamentos que podem ser usados na escolha das características que podem ser analisadas:

1. Para a aplicação em questão é importante o CMS cumprir com suas funcionalidades? Exemplo: O CMS deve possuir boa capacidade para realizar buscas?; O CMS deve ser extensível? etc.
2. Para a aplicação em questão é importante o CMS ter um bom consumo de recursos de hardware, como memória e recursos de entrada e saída?
3. Para a aplicação em questão é importante que o usuário tenha facilidade para desenvolver sua aplicação com CMS?
4. Para a aplicação em questão é importante que o CMS possua uma curva de aprendizado baixa?
5. Para a aplicação em questão é importante que o CMS tenha documentação disponível?
6. Para a aplicação em questão é importante o CMS proteger os dados do usuário e prover um certo nível de restrição de acesso?
7. Para a aplicação em questão é importante CMS possa ser modificado ou mantido por desenvolvedores?
8. Para a aplicação em questão é importante o CMS se adequar a vários browsers diferentes?
9. Para a aplicação em questão é importante que o CMS possua flexibilidade no seu contexto de uso?

10. Para a aplicação em questão é importante que os usuários consigam realizar suas tarefas de forma completa atingindo todos os seus objetivos?

Para saber qual questionamento cada pergunta se refere utilize a Tabela 13. Escolha as características com base nos questionamentos que são mais relevantes para a sua aplicação.

Tabela 13 – Tomada de decisão

Questionamento	Característica
1	<i>Adequação Funcional</i>
2	<i>Eficiência de Desempenho</i>
3	<i>Usabilidade</i>
4	<i>Usabilidade</i>
5	<i>Usabilidade</i>
6	<i>Segurança</i>
7	<i>Manutenabilidade</i>
8	<i>Portabilidade</i>
9	<i>Cobertura de Contexto</i>
10	<i>Efetividade</i>

Passo 2: Medir Característica da SQuaRE candidata.

Baseado nas escolhas do passo 1 desta seção as características escolhidas devem ser submetidas as medições encontradas no apêndice D. Após a execução das métricas deve-se observar os resultados para seguir o fluxograma proposto na imagem 6.3. Caso o resultado da medição associada não seja satisfatório, deve-se voltar ao passo inicial do método na seção 6.4.2.

6.5.3 Objetivo de Medição

Para delimitar o objetivo de medição foram usados os templates definidos nas Tabelas 14 e 15:

Tabela 14 – Objetivo de Medição (SOLLINGEN; BERGHOUT, 1999).

Analisar	O CMS em estudo
Para o propósito de	Conhecer
Com respeito a	Qualidade do Produto segundo a SQuaRE (ISO/IEC 25010,2011)
Do ponto de vista	Do pesquisador

No contexto	De aplicação do CMS em funcionamento num determinado site.
--------------------	--

Tabela 15 – Objetivo de Medição (SOLLINGEN; BERGHOUT, 1999).

Analisar	O CMS em estudo
Para o propósito de	Conhecer
Com respeito a	Qualidade em Uso segundo a SQuaRE (ISO/IEC 25010,2011)
Do ponto de vista	Do pesquisador
No contexto	De aplicação do CMS em funcionamento num determinado site.

6.5.4 Questões

As questões definidas de acordo com o objetivo de medição foram:

1. Qual o percentual (%) de completude funcional existente em um determinado CMS?
2. O usuário consegue realizar tarefas no CMS com sucesso?
3. O CMS em questão possui curva de aprendizagem baixa?
4. A estética (layout) do CMS é proporcionada uma boa experiência para o usuário?
5. O CMS consegue se adaptar a diversos browser, ou sistemas operacionais?
6. Como os componentes que pertencem a um determinado módulo do CMS se relacionam?
7. Os módulos do CMS são reusáveis?
8. O usuário pode modificar um determinado template ou plugin conforme a sua necessidade?
9. O CMS possui algum nível de teste? Exemplos: unidade, aceitação, sistema, integração, regressão.
10. O CMS permite controlar o acesso de seus usuários?
11. O CMS possui proteção contra ataques web? Exemplos: SQL Injection, RFI, Cookie Poisoning?
12. Quanto tempo o CMS leva para carregar uma determinada página?

13. A quantidade de relacionamentos entre os módulos do CMS afeta o desempenho deste CMS?
14. O CMS oferece confiabilidade quando usado?
15. O CMS possui flexibilidade em uso?
16. O CMS é eficaz no seu uso?

6.5.5 Métricas

As métricas definidas de acordo com as questões estão mapeadas no Apêndice D. As métricas estão agrupadas por seções que descrevem as características da SQuaRE mapeadas.

As figuras 37 e 38 apresentam o mapeamento dos objetivos de medição, suas questões e respectivas métricas associadas.

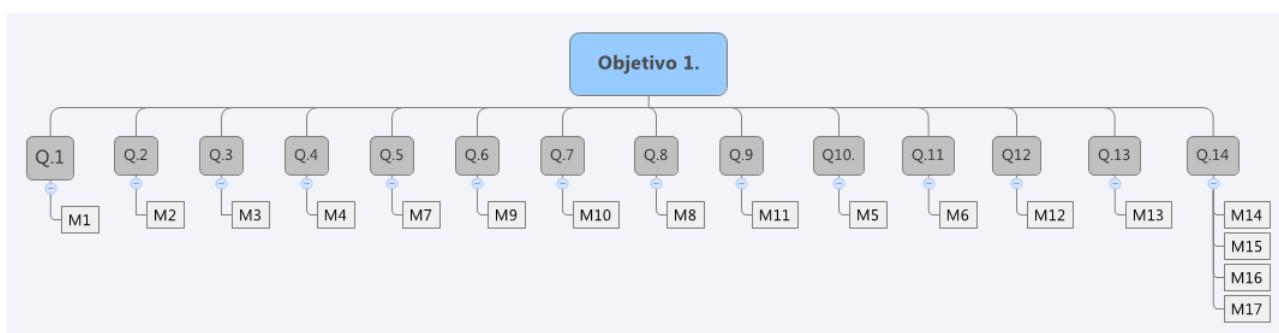


Figura 37 – Diagrama GQM para a Qualidade do Produto



Figura 38 – Diagrama GQM para Qualidade em Uso

6.6 Análise de resultados para o segundo questionário

Para a validação das métricas elaboradas foi construído um novo questionário. Este questionário está descrito na Seção C.2 do Apêndice C. Os procedimentos de aplicação do questionário e o detalhamento da sua estrutura estão presentes nas Seções 2.5.2 e 2.5.3. Esta seção tem como objetivo mostrar o processo de validação do segundo questionário.

6.6.1 Procedimentos necessários para a execução da Análise de Fatores

A análise de fatores tem como principal objetivo expressar um grande número de variáveis em uma quantidade pequena de fatores. A análise de fatores observa padrões de correlações. As correlações representam o relacionamento entre as variáveis envolvidas (DANCEY; REIDY, 2006).

Segundo os mesmos autores, grupos de variáveis altamente correlacionadas entre si formam um fator. Um fator consiste em uma variável hipotética, no qual os seus participantes diferem.

Dancey e Reidy (2006) sugerem que para executar a análise de fatores alguns passos devem ser seguidos, estes passos são:

1. **Produzir a matriz de correlações.**²⁰
2. **Extrair um conjunto de fatores:** O objetivo desta etapa é extrair o mínimo de fatores considerando o máximo de variação. Porém, a decisão sobre o número de fatores a serem extraídos cabe ao pesquisador, que se fundamenta tanto em critérios estatísticos quanto teóricos.
3. **Determinar o número de fatores que deve ser considerado:** uma boa prática para determinar o número de fatores é analisar o gráfico de declividade. O gráfico de declividade relaciona o número de fatores plotados com a quantidade de variância. O gráfico traz a ideia que os fatores decrescem até certo nível e depois se estabilizam formando uma linha horizontal. Como regra observa-se o gráfico e encontra-se o ponto onde o gráfico começa a ficar horizontal. A partir disso, adota-se como fator os pontos demarcados antes ao ponto em que o gráfico começa a se estabilizar. O ponto de estabilização é definido como o ponto com valor próprio igual a 1 (Regra de Kaizer)(MAROCO; GARCIA-MARQUES, 2013).
4. **Verificar as cargas fatoriais:** As cargas fatoriais informam o relacionamento dos itens com os fatores. É uma boa prática encontrar cargas fatoriais de uma matriz

²⁰A matriz de correlações gerada equivale ao conjunto de respostas fornecidos pelo sujeito. Ela é 80 X 19. Em que 80 é a quantidade de respostas fornecidas pelos sujeitos (linhas da matriz) e 19 o número de questões do *survey*(colunas da matriz).

rotacionada. O processo de rotação da matriz pode ser por métodos ortogonais (*Varimax*, *Equamax*), ou por métodos oblíquos (*Direct Oblimin*, *Promax*) (MAROCO; GARCIA-MARQUES, 2013).

As rotações ortogonais são mais fáceis de reportar e de interpretar. Porém, o pesquisador deve assumir que os construtos²¹ são independentes (na prática essa restrição é mais difícil de ser respeitada). Por sua vez, as rotações oblíquas permitem que os fatores sejam correlacionados. Entretanto, são mais difíceis de descrever e interpretar (MAROCO; GARCIA-MARQUES, 2013).

5. **Nomear os fatores encontrados:** Após a rotação, o pesquisador deve observar as cargas fatoriais, a fim de encontrar conjuntos de variáveis comuns. Para isso, a decisão pode ser tomada com base no valor da carga que deve ser incluída. Este processo de escolha do valor da carga base tende a ser arbitrário, porém é de costume escolher uma carga base que varie de 0,3 a 0,5.

6.6.2 Resultados obtidos - Validação da matriz de dados

Antes de iniciar a análise de dados, foi necessária a validação da matriz de dados. Esta validação foi realizada com o auxílio do Software *IBM SPSS Statistics* ®²².

Para a validação da matriz de dados obtida por meio dos resultados fornecidos pelos sujeitos foram usados três índices:

1. **α de Cronbach:** serve para indicar a **confiabilidade** de um dado construto. O índice α estima o quão uniforme os itens contribuem para a soma não ponderada do instrumento, variando numa escala de 0 a 1. Esta propriedade é conhecida por consistência interna da escala e, assim, o α pode ser interpretado como coeficiente médio de todas as estimativas de consistência interna que se obteriam se todas as divisões possíveis da escala fossem feitas (CRONBACH, 1951 apud MAROCO; GARCIA-MARQUES, 2013). Para Nunnally (1978 apud MAROCO; GARCIA-MARQUES, 2013), um instrumento possui boa confiabilidade se o α de Cronbach associado é $\geq 70\%$.
2. **Kaiser-Meyer-Olkin (KMO):** serve para indicar a **consistência dos dados** da matriz em questão. O KMO indica a proporção das variâncias dos dados que podem ser consideradas comuns a todas as variáveis envolvidas. O teste KMO varia entre 0 e 1. Quanto mais perto de 1 melhor, porém para ser aceitável o KMO deve ser $\geq 0,5$ (HAIR et al., 2006 apud FILHO; JÚNIOR, 2010).

²¹Instrumentos de Coleta

²²Disponível em: <http://www-03.ibm.com/software/products/pt/spss-statistics>

3. **Esfericidade de Bartlett (BTS)**: serve para indicar se existem **correlações não nulas** entre as variáveis presentes na matriz de dados. O BTS verifica a hipótese nula de que a matriz de dados é uma matriz identidade. Caso essa hipótese seja rejeitada, então existe consistência entre os dados e a análise fatorial pode ser aplicada (FERREIRA; BAPTISTA; ., 2004 apud CRUZ; TOPA, 2009). O teste BTS para ser aceitável deve ser $\leq 0,05$ (HAIR et al., 2006 apud FILHO; JÚNIOR, 2010).

Após a submissão da matriz de dados no Software SPSS obteve-se como resultados para os indicadores, os dados mostrados na tabela 16:

Tabela 16 – Resultados para os indicadores estabelecidos

	Critério de aceitação	Resultado obtido
α	$\geq 0,7$	0,746
KMO	$\geq 0,5$	0,636
BTS	$\leq 0,05$	0,000

A partir dos dados apresentados conclui-se que a matriz de dados é válida e que pode ser fatorada, isto é, reduzida em fatores.

6.6.3 Extração de fatores

Para a encontrar a quantidade de fatores existentes na matriz de dados, foi gerado um gráfico de declividade. A Figura 39 mostra o cruzamento dos valores próprios²³ com o número de questões existentes.

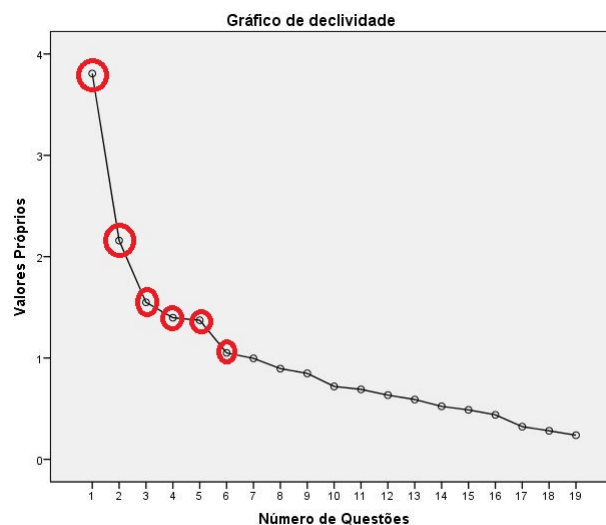


Figura 39 – Gráfico de declividade para a matriz de dados

²³Número que representa o quanto um fator é relevante para o conceito.

Os pontos circulados mostram as declividades que representam os fatores. Após o par de pontos (7,1), isto é, após a questão 7 ocorre a estabilização do gráfico, onde o valor próprio já está abaixo de 1. O gráfico sugere seis fatores nos quais as questões elaboradas podem ser agrupadas.

Foi escolhida a rotação oblíqua *Direct Oblimin*, porque de acordo com os autor (DAMÁSIO, 2012) ela busca uma solução simples com maior correlação entre os fatores envolvidos.

Quanto ao valor mínimo de aceitação para as cargas de fatores, foi fixado o valor de 0,3; valor semelhante ao usado por Maroco e Garcia-Marques (2013) e dentro do intervalo sugerido por Dancey e Reidy (2006) em suas pesquisas.

Do ponto de vista teórico o agrupamento de questões para 6 fatores como indicado na Tabela 50 do Apêndice F e da Figura 39, não apresenta nenhuma semelhança do ponto de vista da literatura SQuaRE. Isto é, tomando como exemplo o Fator 2, que é composto pelos conceitos Confidencialidade, Disponibilidade e Tolerância a Falhas, observa-se que estes conceitos pertencem a características diferentes da SQuaRE (Segurança e Confiabilidade). Estes conceitos não se relacionam. Com esta motivação, foram feitas reduções no número de fatores buscando alguma semelhança com a SQuaRE. Além disso, o gráfico de declividade mostra que os fatores possuem um valor próprio bastante próximo, o que indica que os fatores podem ser reduzidos.

A título de exercício foram citadas outras soluções fatoriais, com cinco, quatro, três e dois fatores. As Tabelas 51, 52, 53, 54 do Apêndice F exemplificam estas reduções. No entanto, a solução fatorial que mais se aproximou da literatura foi a de 1 fator, como mostra a Tabela 17.

Tabela 17 – Cargas fatoriais para um fator

Matriz de componentes - um fator	
Questão - Conceito	Componente 1
Q1 - Completude Funcional	0,237
Q2 - Operacionalidade	0,340
Q3 - Aprendizado	0,265
Q4 - Estética	0,662
Q5 - Confidencialidade	0,075
Q6 - Integridade	0,090
Q7- Adaptabilidade	0,353
Q8 - Modificabilidade	0,389
Q9 - Reusabilidade	0,423
Q10 - Modularidade	0,539

Q11 - Testabilidade	0,640
Q12 - Comportamento do Tempo	0,652
Q13 - Utilização de Recursos	0,548
Q14 - Maturidade	0,476
Q15 - Disponibilidade	0,279
Q16 - Tolerância a Falhas	0,377
Q17 - Recuperabilidade	0,702
Q18 - Efetividade	0,410
Q19 - Flexibilidade	0,355

Foi observado que alguns conceitos não possuem carga fatorial suficiente. Considerando o contexto do trabalho elaborado, esta divisão não apresenta restrições quanto a sua organização, logo todos os conceitos presentes no Fator 1 podem ser chamados de *métricas mais significativas* para os especialistas. Conclui-se, então, que as métricas mais significativas para o estudo são as que possuem carga fatorial $\geq 0,3$ no Fator 1.

Porém, ainda podem ser percebidas algumas métricas menos significantes, isto é possuem carga fatorial abaixo do estabelecido. Observando as métricas excluídas, nota-se que alguns dos conceitos que foram considerados importantes no decorrer do trabalho como a Adequação Funcional (Completo Funcional) e a Segurança (Confidencialidade, Integridade), ficaram de fora do fator por não terem carga fatorial suficiente.

A partir das métricas com carga fatorial insignificante foi feita a média e o desvio padrão, a fim de estabelecer uma análise mais completa dos dados. A Tabela 18 mostra esses dados.

Tabela 18 – Média e Desvio Padrão para questões com carga fatorial insuficiente.

	Q1	Q3	Q5	Q6	Q15
Média	3,688	3,625	3,888	3,900	3,675
Desvio Padrão	,4928	,5819	,3180	,3019	,5905

A partir da análise dos dados presentes na Tabela 18, é possível concluir que:

- A média para as questões permanece em volta da escala Muito Importante e Extremamente Importante.
- O desvio padrão está baixo em relação a média, o que mostra que os dados levantados para as questões possuem baixa dispersão.

Por terem baixa dispersão e assim estarem bem próximos da média, as métricas com carga fatorial baixa não podem ser completamente descartadas. Apesar da análise

de fatores ter reduzido o número de métricas, a aplicação do método deve ser reaplicada considerando um aumento no número de sujeitos. Além disso, pode - se constatar que o fato de tais conceitos serem indispensáveis para um CMS pode ter ocasionado a exclusão destes conceitos pelo método da Análise de fatores.

6.6.4 Breve estatística descritiva para as Métricas mais significantes

As Figuras de 40 a 53 mostram as frequências de respostas dos sujeitos entrevistados para as questões mapeadas na seção 6.6.3.

Para a leitura das figuras considere a seguinte legenda:

1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante.

2) No que diz respeito a Usabilidade complete: as mensagens exibidas pelo CMS são_____.

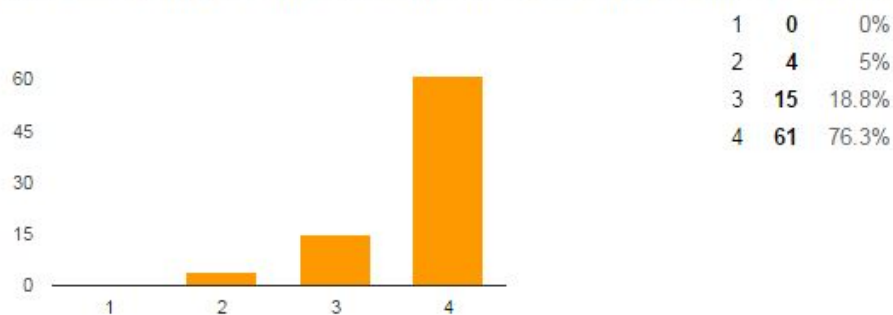


Figura 40 – Frequência de respostas para a métrica de operacionalidade

4) No que diz respeito a Usabilidade complete: o layout é _____ em um determinado CMS.

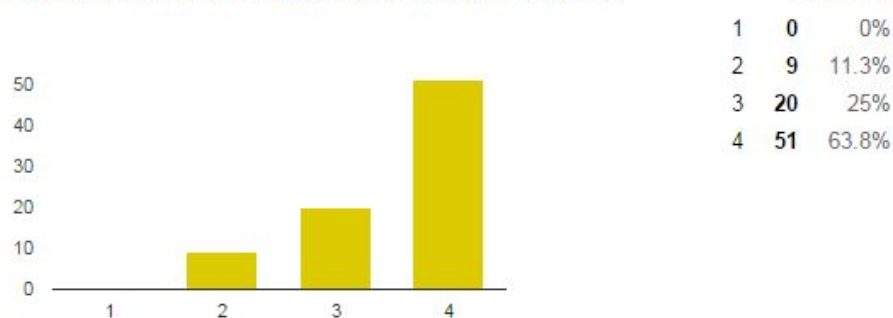


Figura 41 – Frequência de respostas para a métrica de estética

8) No que diz respeito a Manutenibilidade complete: modificar um CMS para resolver um determinado problema é ...

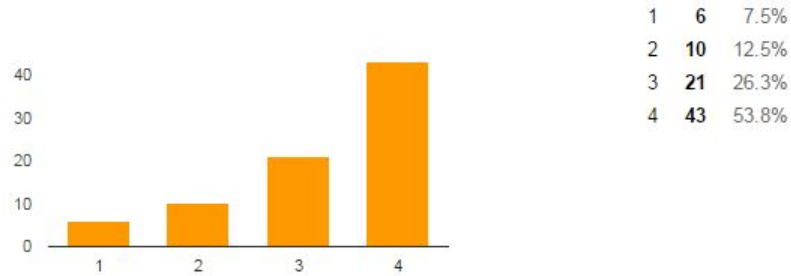


Figura 42 – Frequência de respostas para a métrica de adaptabilidade

8) No que diz respeito a Manutenibilidade complete: modificar um CMS para resolver um determinado problema é ...



Figura 43 – Frequência de respostas para a métrica de modificabilidade

9) No que diz respeito a Manutenibilidade complete. É _____ o CMS possuir módulos reusáveis.

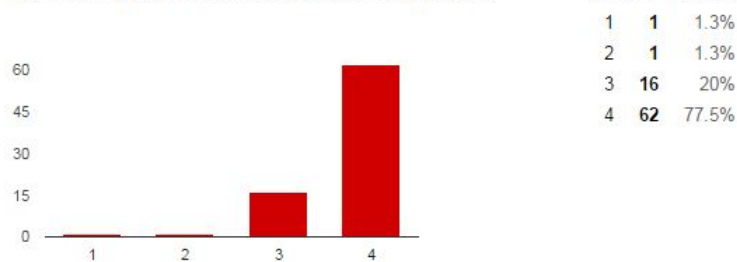


Figura 44 – Frequência de respostas para a métrica de reusabilidade

10) No que diz respeito a Manutenibilidade complete: o relacionamento entre os componentes é _____ para o funcionamento do módulo.

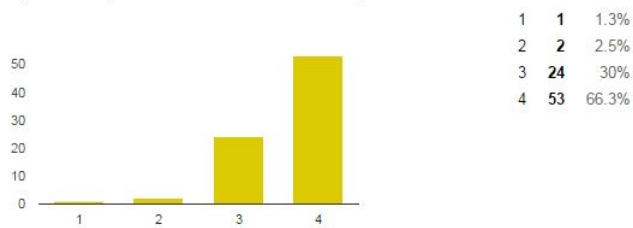


Figura 45 – Frequência de respostas para a métrica de modularidade

11) No que diz respeito a Manutenibilidade complete : testes de sistema são _____ para um CMS.

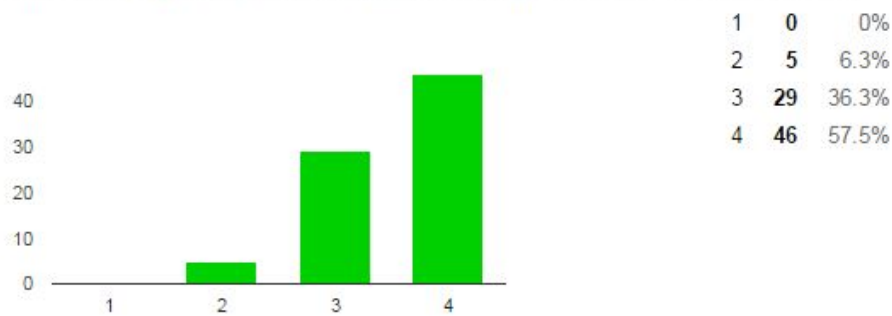


Figura 46 – Frequência de respostas para a métrica de testabilidade

12) No que diz respeito a Performance complete : a quantidade de tarefas que pode ser realizada por período de tempo é _____.



Figura 47 – Frequência de respostas para a métrica de comportamento do tempo

13) No que diz respeito a Performance complete: a utilização de memória e outros recursos de hardware é _____ em um CMS.

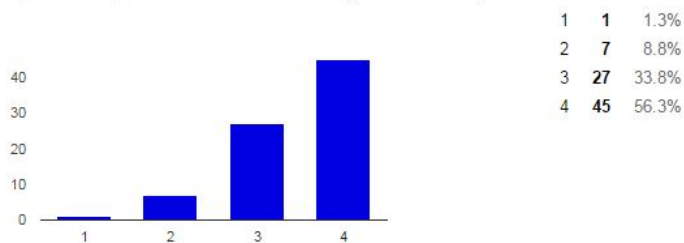


Figura 48 – Frequência de respostas para a métrica de Utilização de Recursos

14) No que diz respeito a confiabilidade completa: a capacidade de falhas serem identificadas e corrigidas em um projeto com uso de CMS é _____.



Figura 49 – Frequência de respostas para a métrica de maturidade

14) No que diz respeito a confiabilidade completa: a capacidade de falhas serem identificadas e corrigidas em um projeto com uso de CMS é _____.



Figura 50 – Frequência de respostas para a métrica de tolerância a falhas

17) No que diz respeito a Confiabilidade completa: é _____ o CMS se recuperar após uma falha.

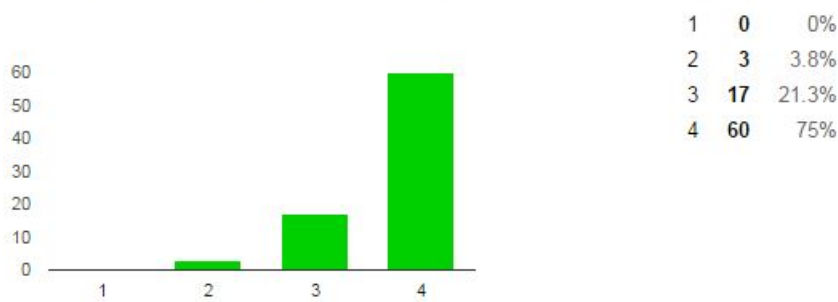


Figura 51 – Frequência de respostas para a métrica de recuperabilidade

18) No que diz respeito a Efetividade completa: é _____ um usuário realizar suas atividades e conseguir completa-las de forma correta.

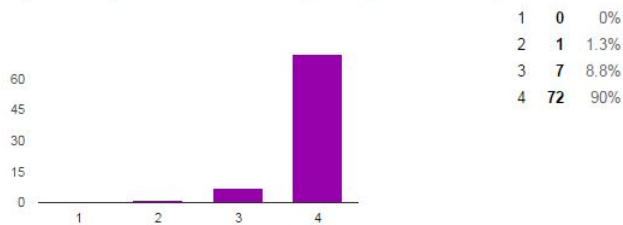


Figura 52 – Frequência de respostas para a métrica de efetividade

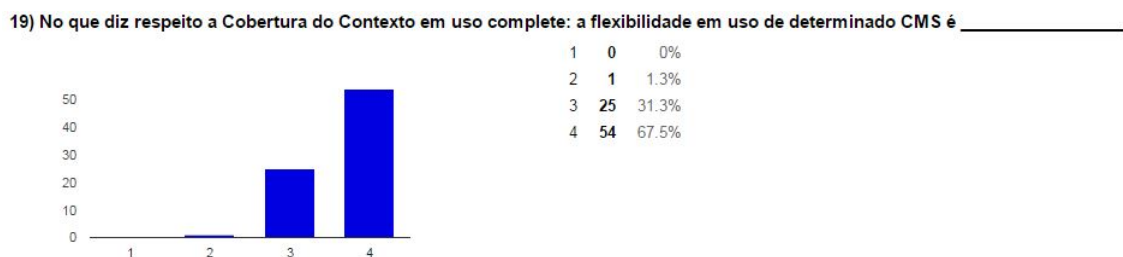


Figura 53 – Frequência de respostas para a métrica de flexibilidade

As Tabelas 19 e 20 apresentam um resumo da média, moda, mediana e desvio padrão para as características mais importantes.

Tabela 19 – Outras medidas para as métricas mais relevantes

	Q2	Q4	Q7	Q8	Q9	Q10	Q11	Q12
Média	3,7125	3,5250	3,1000	3,2625	3,7375	3,6125	3,5125	3,5000
Mediana	4,0000	4,0000	3,0000	4,0000	4,0000	4,0000	4,0000	4,0000
Moda	4,0000	4,0000	3,0000	4,0000	4,0000	4,0000	4,0000	4,0000
Desvio padrão	0,5556	0,6931	0,8509	0,9513	0,5453	0,6057	0,6161	0,6560

Tabela 20 – Outras medidas para as métricas mais relevantes

	Q13	Q14	Q16	Q17	Q18	Q19
Média	3,4500	3,8375	3,4125	3,7125	3,8875	3,6625
Mediana	4,0000	4,0000	4,0000	4,0000	4,0000	4,0000
Moda	4,0000	4,0000	4,0000	4,0000	4,0000	4,0000
Desvio padrão	0,7098	0,3712	0,7745	0,5323	0,3556	0,5017

Observa-se que os desvios padrões para as questões elaboradas também são baixos, o que mostra baixa discrepância em relação a média calculada. Alguns dos fatores pelos quais podem justificar essa baixa são:

- Poucos sujeitos;
- Grupo de sujeitos homogêneo, ou seja, as pessoas possuem a mesma perspectiva sobre as métricas levantadas. Fato que pode ser justificado pelo fato do questionário 2 ser um refinamento de características já mapeadas no questionário anterior.

A partir da observação dos resultados fornecidos pela análise de fatores e da estatística descritiva foi constatado que não existem métricas mais relevantes, ou menos relevantes, pois todas as métricas envolvidas possuem média alta e fazem parte de um conceito geral chamado "Qualidade de Software" de acordo com a percepção do sujeito.

As métricas descartadas pela análise de fatores não podem ser completamente desconsideradas por terem representatividade suficiente na literatura, porém o fato de terem sido excluídas da análise de fatores aponta que tais características associadas sejam indispensáveis em um CMS, fato que pode ser experimentado por meio de trabalhos futuros com segregação e aumento no número da amostra de sujeitos.

Com as constatações feitas a partir da análise de fatores o método foi concluído. O método é bom, pois cumpre com aquilo que se propõe a fazer que é selecionar um CMS baseado em características de qualidade. Alguns refinamentos ainda podem ser feitos. Estes refinamentos estão detalhados na seção [7.1](#).

7 Conclusões

O uso de CMS para o desenvolvimento de páginas web permite ao desenvolvedor publicar e editar conteúdos sem um conhecimento pleno de linguagens de programação para web. Além disso, o desenvolvedor possui a liberdade de adicionar novas funcionalidades, ou mudar o design da sua página por meio do uso de plugins e templates.

Por serem boas alternativas para o desenvolvimento web, a quantidade de CMSs disponíveis no mercado é bastante significativa. Devido ao grande número de CMSs disponíveis, escolher a melhor opção para um determinado contexto pode não ser algo simples. Contextos que podem variar desde aplicações para e-commerce, redes sociais, revistas eletrônicas, blogs, sites para instituições bancárias e outros, são contextos usados com frequência por quem utiliza CMS.

Em outras palavras, a escolha de um CMS pode influenciar várias características funcionais, ou não funcionais de uma determinada aplicação web. A partir desta motivação, o estudo de critérios que facilitem a escolha de produtos do tipo CMS foi importante para esta pesquisa.

Motivação que serviu como base para a seguinte questão de pesquisa:

"Como estabelecer um método objetivo para a escolha de CMSs a serem utilizados no desenvolvimento de sites Web?"

Questão de pesquisa que fundamentou-se no seguinte objetivo geral:

"Propor um método objetivo que possa servir de apoio para a escolha de um CMS."

Para atender ao objetivo geral foram definidos seis objetivos específicos, que foram resolvidos ao longo do trabalho de conclusão de curso.

Para o objetivo específico número um "*Identificar o que é um CMS*" foi realizado um estudo a respeito do que é CMS, quais as vantagens e desvantagens do uso de CMS para o desenvolvimento web, algumas aplicações feitas com auxílio de CMSs. Após esse estudo foi definido dois critérios importantes para o estudo: o licenciamento, que compreende o fato de um CMS ser software livre ou proprietário e a popularidade. Essas definições de estudo estão presentes no Capítulo 3 deste trabalho.

Para o objetivo específico dois "*Examinar conceitos de qualidade de software, a fim de esclarecer como medir as características de sistemas CMSs, a fim de permitir que elas sejam mensuradas*" foi realizado um estudo a respeito da ISO/IEC 25000, série SQuaRE de normas. Neste estudo foi abordado a sua organização, a sua visão de qualidade em uso

e qualidade do produto e um comparativo com a antiga ISO/IEC 9126. Essas definições de estudo estão presentes no Capítulo 4 deste trabalho.

Para o objetivo específico três "*Definir medições para avaliar as características de CMSs a partir de conceitos das teorias para estabelecer medições usadas na Engenharia de Software*" foi realizado um estudo sobre tipos de métricas de software, sobre escalas de software e sobre métodos para a elaboração de métricas de software. Métodos que compreendem o *Goal, Question, Metric-GQM* e o *Practical Software Measurement-PSM*. Este estudo foi abordado no Capítulo 5.

Para o objetivo específico quatro "*Identificar sistemas de CMSs candidatos para serem usados como estudos de caso na aplicação do método proposto*" foi realizado um estudo a respeito do licenciamento e da popularidade em sistemas CMS. Durante esse estudo foram citados diversos CMSs, porém com a aplicação dos critérios de software livre, popularidade e tamanho de comunidade. O tamanho da comunidade foi medido por meio de redes sociais e o número de commits no github. A partir dos filtros aplicados o número de CMSs se reduziu aos mais populares: Joomla, Wordpress e Drupal. Este estudo foi abordado na Seção 3.4 do Capítulo 3.

Para o objetivo específico cinco "*Estabelecer um método que possa ser aplicado aos CMSs candidatos, a partir de medições oriundas de conceitos de qualidade de software*" foi realizado um levantamento de características de CMS com base na literatura. As características citadas na literatura foram mapeadas com suas respectivas características SQuaRE. Este mapeamento teve como objetivo agrupar as características semelhantes e mais citadas para a elaboração de um *survey*. *Survey* que foi submetido e respondido por especialistas para elencar as características de CMS mais relevantes.

Após o estudo do *survey* as características mais relevantes foram agrupadas em dois grupos. O grupo um foi composto por características que não podem ser mensuradas pela Square, já o dois pelas que podem. A partir disso, foi definida uma visão geral para o método (Seção 6.3) que foi dividido em duas partes quanto a sua execução. Para a 2ª parte do método foi feito um GQM com a finalidade de definir as métricas necessárias para o estudo. Estas definições de estudo foram apresentadas no Capítulo 6.

Para o último objetivo específico "*Validar o método proposto a partir de opiniões de especialistas no assunto*" foi feito um segundo *survey* com o objetivo de validar as métricas elaboradas para o método proposto. Os resultados para este *survey* foram citados na Seção 6.6. Para a validação do *survey* foi aplicado o método da análise de fatores. Para prosseguir com o método da análise de fatores foi necessário produzir uma matriz de dados e verificar a validade da mesma. Para aferir a validade foram utilizados três indicadores. Esses indicadores foram: o α de Cronbach, o KMO e o BTS.

Depois da validação da matriz de dados elaborada constatou-se que a mesma

podia ser quebrada em fatores. A partir da validação foi iniciada a extração de fatores com rotação da matriz. Foi gerado um gráfico de declividade, no qual foram sugeridos seis fatores. Inicialmente os seis fatores foram observados, porém verificou-se a necessidade da redução em menos fatores, pois os conceitos associados aos fatores não eram compatíveis com a literatura.

As reduções foram feitas até se chegar a um único fator. Este fator foi nomeado de métricas mais significativas. Porém, foi constatado que conceitos importantes para a literatura não foram considerados pela análise de fatores. Estes conceitos foram a Segurança e a Adequação Funcional.

A causa para a desconsideração desses conceitos deve ser atribuída ao baixo número de sujeitos. Foram 80 sujeitos para 19 questões mapeadas, o que leva a uma média de 4 sujeitos por item.

A partir do método da análise de fatores foi percebido que um passo crítico para o método proposto é a escolha de características a partir dos critérios que dizem respeito ao tipo de site que será produzido, uma vez que, as métricas selecionadas representam o conceito geral de Qualidade de Software para a opinião dos sujeitos entrevistados. Os critérios apresentados para a escolha de CMSs de acordo com a aplicação a ser desenvolvida estão na Seção 6.5.2 deste trabalho.

Após a avaliação do método por especialistas concluiu-se o método. A grande contribuição do método para a indústria de software foi que com a sua utilização é possível o desenvolvedor testar se o CMS é adequado a uma determinada característica. Característica que caso venha a ser importante para uma determinada aplicação pode ser avaliada por meio de métricas estabelecidas no método, e então, avaliar se o CMS é adequado para a aplicação em questão.

7.1 Sugestões de Trabalhos Futuros

Apesar do método estar de acordo com a visão dos especialistas, conforme foi mostrado nos questionários aplicados. A visão dos especialistas encontrados reflete apenas a visão de quem usa, ou usou um CMS software livre para os seus projetos. A visão de especialistas ligados ao uso de CMSs proprietários poderia também ser abordada. Para isso poderiam ser realizadas entrevistas com desenvolvedores que trabalham com CMSs proprietários em um determinado ambiente organizacional, afim de colher opiniões e obter uma amostra mais refinada de características para refinar o método proposto.

Uma validação por experimentos também poderia ser executada. Para esta validação poderia ser escolhido um número "X" de CMSs, e em seguida os CMSs escolhidos seriam submetidos aos passos descritos nas Seções 6.4.2 e 6.5. Durante o experimento

o foco poderia ser a análise de uma característica como por exemplo, a Usabilidade. A partir da aplicação deste experimento poderia ser verificado se os CMSs "X" escolhidos atenderiam a característica de Usabilidade por meio das métricas propostas.

As metas para as medições propostas são apenas sugestões. Estas metas poderiam ser verificadas propondo-se vários experimentos quantitativos com a aplicação das próprias métricas sugeridas envolvendo um conjunto de CMSs escolhidos.

Devido ao baixo número de sujeitos (4/item) a reaplicação do método de análise de fatores deve ser considerada. Um bom número de sujeitos seria (10/item), o que equivaleria a 190 sujeitos. A partir disso, também seria necessária uma reaplicação do questionário de validação de métricas, considerando uma validação semântica do questionário a fim de melhorar o instrumento de coleta e possibilitar melhores resultados.

Apesar de conceitos importantes terem sido desconsiderados, no resultado final da análise de fatores, as métricas mais significativas representam o conceito geral de qualidade de software para a opinião dos especialistas entrevistados. Opções que poderiam ser agrupadas quanto a experiência (desenvolvedor, usuário, mantenedor) do especialista em relação ao uso de CMS. Possibilitando uma nova sugestão para reaplicação do método de análise de fatores, a fim de perceber se a acurácia dos fatores seria melhorada.

Além disso, uma investigação mais profunda dos conceitos indispensáveis para sistemas CMSs se faz necessária devido aos resultados encontrados com a análise de fatores.

Referências

- ADHONI, Z.; HAMAD, H. A. An API for Quran portal using Drupal technology. In: *Applications of Digital Information and Web Technologies (ICADIWT), 2014 Fifth International Conference on the*. [S.l.: s.n.], 2014. p. 160–164. Citado na página 121.
- AGUIAR, T. et al. Anati quanti: Software de análises quantitativas para estudos em anatomia vegetal. *Planta daninha*, v. 25, n. 4, p. 649–659, 2007. Citado na página 35.
- ALBRECHT, C.; MATTHEWS, K. Learning 2.0 in the Information Systems Curriculum. In: *42nd Hawaii International Conference on System Sciences, 2009. HICSS '09*. [S.l.: s.n.], 2009. p. 1–10. Citado na página 122.
- ALVES, P.; PIRES, J. A. A usabilidade em software educativo: princípios e técnicas. 2002. Citado na página 160.
- AW, S. L. et al. Integration of captivate and JOOMLA: Online demonstration and assessment eLearning resources. In: *Electronics and Information Engineering (ICEIE), 2010 International Conference On*. [S.l.: s.n.], 2010. v. 1, p. V1–137–V1–140. Citado na página 121.
- BAILEY, E. et al. Practical software and systems measurement. *DoD and US Army*, 2003. Citado 2 vezes nas páginas 13 e 61.
- BALASKO, A. et al. Supporting Agricultural Communities with Workflows on Heterogeneous Computing Resources. In: *2014 6th International Workshop on Science Gateways (IWSG)*. [S.l.: s.n.], 2014. p. 18–23. Citado na página 122.
- BARCIA, L. M. G. *A utilização da plataforma joomla! na escola*. Dissertação ((Mestrado em Ciências da Educação)) — Universidade Católica Portuguesa, 2008. Citado na página 21.
- BARRÉRE; AVELINO, I.; EDUARDO, J. K. Construção de sites para comunidades virtuais e intranet utilizando cms. S/D. Citado 4 vezes nas páginas 21, 31, 33 e 34.
- BASIL, V. R.; CALDIERA, G.; ROMBACH, H. D. The goal question metric approach. 1994. Citado 3 vezes nas páginas 11, 59 e 60.
- BORGES, E. P. *Um modelo de medição para processos de desenvolvimento de software*. Dissertação (Mestrado.) — Universidade Federal de Minas Gerais - UFMG. Curso de Pós-graduação em Ciência da Computação, Belo Horizonte - MG, Março 2003. Citado 3 vezes nas páginas 11, 60 e 63.
- BORTOLI, L. A. D. Um método de trabalho para auxiliar a definição de requisitos. 1999. Citado 2 vezes nas páginas 32 e 33.
- BOUKAR, M.; MUSLU, I. Administration and academic staff performance management system using content management system (cms) technologies. In: *Electronics, Computer and Computation (ICECCO), 2013 International Conference on*. [S.l.: s.n.], 2013. p. 151–154. Citado na página 32.

- BURGESS, A.; MATTMANN, C. Automatically classifying and interpreting polar datasets with Apache Tika. In: *2014 IEEE 15th International Conference on Information Reuse and Integration (IRI)*. [S.l.: s.n.], 2014. p. 863–867. Citado na página 122.
- CARD, D. N. Practical software measurement. In: *Software Engineering, 2003. Proceedings. 25th International Conference on*. [S.l.: s.n.], 2003. p. 738–739. ISSN 0270-5257. Citado 3 vezes nas páginas 58, 59 e 64.
- CARD, D. N.; JONES, C. Status report: practical software measurement. In: *Quality Software, 2003. Proceedings. Third International Conference on*. [S.l.: s.n.], 2003. p. 315–320. Citado 5 vezes nas páginas 11, 60, 61, 62 e 64.
- CHENG, X.; WANG, Z. The Web Development Based on the Drupal System. In: *2012 Second International Conference on Business Computing and Global Informatization (BCGIN)*. [S.l.: s.n.], 2012. p. 778–780. Citado 3 vezes nas páginas 67, 120 e 128.
- CMSMATRIX: Vignette web content management (now opentext web experience management). 2014. Disponível em: <<http://www.cmsmatrix.org/matrix/cms-matrix/vignette-web-content-management-now-opentext-web-experience-management->>. Acesso em: 03 Nov. 2014. Citado na página 35.
- COELHO, M. A. P. et al. O uso do cms joomla e suas ferramentas hipertextuais na produção de sites educativos e de material didático online. *Texto Livre: Linguagem e Tecnologia*, v. 4, n. 2, 2011. Citado na página 21.
- CONALLEN, J. *Desenvolvendo aplicações web*. [S.l.]: –, 2003. Citado na página 21.
- COREMEDIA: Site. 2014. Disponível em: <<http://www.coremedia.com/>>. Acesso em: 03 Nov. 2014. Citado na página 35.
- COSTA, L. N. da; CAMINHA, H. E. B.; LIMA, W. J. F. de. Comparativo de ferramentas de criação e gestão de web sites cms (content management system). *Sistemas de Informação & Gestão de Tecnologia.*, No 6 (2012), 2012. Citado 7 vezes nas páginas 13, 23, 33, 41, 42, 119 e 125.
- CRONBACH, L. *Coefficient alpha and the internal structure of tests*. [S.l.: s.n.], 1951. Citado na página 91.
- CRUZ, I. C. da; TOPA, M. A. *ANÁLISE MULTIVARIADA COMO FERRAMENTA DE GERENCIAMENTO DE FORNECEDORES VISANDO UM RELACIONAMENTO COM VANTAGEM COMPETITIVA*. Dissertação (Monografia) — Universidade Federal Do Paraná - Bacharelado em Estatística, 2009. Citado na página 92.
- CYMBALAK, D.; JAKAB, F.; MICHALKO, M. Advanced solution for delivering educational multimedia content based on content management system. In: *2012 IEEE 10th International Conference on Emerging eLearning Technologies Applications (ICETA)*. [S.l.: s.n.], 2012. p. 67–72. Citado 2 vezes nas páginas 122 e 130.
- CYSNEIROS, L. M. *Requisitos Não Funcionais: Da Elicitação ao Modelo Conceitual*. Tese (Doutorado) — DEPARTAMENTO DE INFORMÁTICA -PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, 2001. Citado na página 32.
- CYSNEIROS, L. M.; LEITE, J. Definindo requisitos não funcionais. *XI Simpósio Brasileiro de Engenharia de Software. Fortaleza, CE*, 1997. Citado na página 33.

DAMÁSIO, B. F. Uso da análise fatorial exploratória em psicologia. *Avaliação Psicológica*, scielopepsic, v. 11, p. 213 – 228, 08 2012. ISSN 1677-0471. Disponível em: <http://pepsic.bvsalud.org/scielo.php?script=sci_arttext&pid=S1677-04712012000200007&nrm=iso>. Citado na página 93.

DANCEY, C. P.; REIDY, J. *Estática sem Matemática para Psicologia - Usando o SPSS para Windows*. 3ª edição. ed. [S.l.]: tradução Lorí Viali. - Porto Alegre:, 2006. Citado 2 vezes nas páginas 90 e 93.

DISTANTE, D.; RISI, M.; SCANNIELLO, G. Extending web content management systems navigation capabilities with semantic navigation maps. In: *2010 12th IEEE International Symposium on Web Systems Evolution (WSE)*. [S.l.: s.n.], 2010. p. 1–5. Citado 3 vezes nas páginas 121, 125 e 129.

DOROSH, O.; KUCHMIJ, N. Designing of e-commerce system by CMS Joomla software. In: *10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics, 2009. CADSM 2009*. [S.l.: s.n.], 2009. p. 400–400. Citado na página 120.

DUCHESSI, P.; CHENGALUR-SMITH, I. Client/server benefits, problems, best practices. *Commun. ACM*, ACM, New York, NY, USA, v. 41, n. 5, p. 87–94, maio 1998. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/274946.274961>>. Citado na página 20.

ENNERT, M.; CAJKOVSKY, M.; ADAM, N. Optimization model of management and security of multimedia content using Drupal CMS. In: *2012 IEEE 10th International Symposium on Applied Machine Intelligence and Informatics (SAMi)*. [S.l.: s.n.], 2012. p. 289–293. Citado 4 vezes nas páginas 120, 126, 129 e 130.

FACEBOOK: Site. 2014. Disponível em: <<https://pt-br.facebook.com/>>. Acesso em: 05 Out. 2014. Citado na página 37.

FENTON, N. E.; PFLEGEER, S. L. *Software metrics – A rigorous approach*. [S.l.]: Publishing Company, 1997. Citado 4 vezes nas páginas 13, 55, 56 e 57.

FERREIRA, J. S.; BAPTISTA, A. J. M. S.; ., J. E. L. A modernização agropecuária nas microrregiões do estado de minas gerais. *Revista de Economia e Sociologia Rural, Brasília*, 2004. Citado na página 92.

FILHO, D. B. F.; JÚNIOR, J. A. d. S. Visão além do alcance: uma introdução à análise fatorial. *Opinião Pública*, Universidade Estadual de Campinas, v. 16, n. 1, p. 160–185, 2010. Citado 2 vezes nas páginas 91 e 92.

FILHO, W. de P. P. Engenharia de software. LTC, 2003. Citado 2 vezes nas páginas 23 e 32.

GAJDKA, K.; ZABIEROWSKI, W.; NAPIERALSKI, A. Building computer system for musical school with CSM Plone. In: *2008 Proceedings of International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science*. [S.l.: s.n.], 2008. p. 582–584. Citado 3 vezes nas páginas 119, 126 e 127.

- GALSTER, M.; TOFAN, D. Exploring possibilities to analyse microblogs for dependability information in variability-intensive open source software systems. In: *2013 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW)*. [S.l.: s.n.], 2013. p. 321–325. Citado 3 vezes nas páginas 121, 129 e 130.
- GIANNOPOULOU, E. et al. A Semantic Web Approach in the Implementation of a Linked Data Portal Using a CMS. In: *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems (SITIS)*. [S.l.: s.n.], 2014. p. 164–171. Citado na página 122.
- GIOUVANAKIS, E. et al. A game with a purpose for annotating Greek folk music in a web content management system. In: *2013 18th International Conference on Digital Signal Processing (DSP)*. [S.l.: s.n.], 2013. p. 1–6. Citado na página 120.
- GITHUB: Site. 2014. Disponível em: <<https://github.com/>>. Acesso em: 02 Set. 2014. Citado na página 37.
- HAIR, J. et al. *Multivariate Data Analysis*. [S.l.]: Upper Saddle River, NJ, 2006. Citado 2 vezes nas páginas 91 e 92.
- HAMID, N.; HASAN, M. Industrial-based object-oriented software quality measurement system and its importance. In: *Information Technology (ITSim), 2010 International Symposium in*. [S.l.: s.n.], 2010. v. 3, p. 1332–1336. ISSN 2155-897. Citado na página 46.
- HANLON, M. et al. My-Plant.org: A phylogenetically structured social network. In: *Gateway Computing Environments Workshop (GCE), 2010*. [S.l.: s.n.], 2010. p. 1–8. Citado 2 vezes nas páginas 122 e 130.
- HETKA, P.; ZABIEROWSKI, W.; NAPIERALSKI, A. The development of a new branch of the economy: e-commerce. Practical use with Content Management Systems Joomla! and enlargement VirtueMart. In: *10th International Conference - The Experience of Designing and Application of CAD Systems in Microelectronics, 2009. CADSM 2009*. [S.l.: s.n.], 2009. p. 413–419. Citado na página 121.
- HOCHBERGER, C.; MEUSEL, C. Infrastructure for web-based administration of embedded systems. In: *Industrial Embedded Systems, 2008. SIES 2008. International Symposium on*. [S.l.: s.n.], 2008. p. 67–73. Citado na página 21.
- ILIEV, I.; VASILEVA, S. Realization of electronic textbook by means of Drupal Content Management System. In: *2013 Second International Conference on e-Learning and e-Technologies in Education (ICEEE)*. [S.l.: s.n.], 2013. p. 266–272. Citado 4 vezes nas páginas 121, 125, 128 e 129.
- IMPERVA. *Cookie Poisoning*. 2015. Disponível em: <http://www.imperva.com/Resources/Glossary?term=cookie_poisoning>. Acesso em: 13 May. 2015. Citado na página 128.
- ISO/IEC-15939. *Software Engineering - Software Measurement Process*. [S.l.]: ISO/IEC, 2007. Citado na página 47.
- ISO/IEC-25000. *Software engineering — Software product Quality Requirements and Evaluation (SQuaRE) — Guide to SQuaRE*. [S.l.]: ISO/IEC, 2005. Citado 6 vezes nas páginas 11, 23, 45, 47, 48 e 68.

- ISO/IEC-25010. *Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — System and software quality models*. [S.l.]: ISO/IEC, 2011. Citado 9 vezes nas páginas 11, 13, 46, 48, 49, 50, 51, 52 e 53.
- ISO/IEC-25022. *Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of quality in use*. [S.l.]: ISO/IEC, 2012. Citado 4 vezes nas páginas 14, 85, 157 e 158.
- ISO/IEC-25023. *Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – Measurement of system and software product quality*. [S.l.]: ISO/IEC, 2011. Citado 13 vezes nas páginas 13, 14, 83, 148, 149, 150, 151, 152, 153, 154, 155, 156 e 157.
- ISO/IEC-9126-1. *ISO/IEC 9126. Software engineering – Product quality*. [S.l.]: ISO/IEC, 2001. Citado na página 53.
- ISO/IEC-9126-2. *ISO/IEC 9126. Software engineering – Product quality*. [S.l.]: ISO/IEC, 2001. Citado 2 vezes nas páginas 14 e 151.
- JINWEI, F.; JIANHONG, S.; HAICHENG, X. A Multi-sites Scheme Based on Open Source CMS, Drupal. In: *2010 Second International Conference on Multimedia and Information Technology (MMIT)*. [S.l.: s.n.], 2010. v. 2, p. 239–241. Citado na página 121.
- JIUGEN, Y.; RUONAN, X.; QI, W. Construction of education application platform based on Plone content management system. In: *2012 7th International Conference on Computer Science Education (ICCSE)*. [S.l.: s.n.], 2012. p. 513–516. Citado na página 123.
- JONES, C. *Applied Software Measurement: Global Analysis of Productivity and Quality*. 3rd. ed. [S.l.]: New York:, 2008. Citado na página 125.
- JONES, K. M. L.; FARRINGTON, P.-A. WordPress as library CMS: free and open source technology to enhance your library’s web presence.(Dispatches from the Field). *American Libraries*, v. 42, n. 5 6, p. 34, 2011. ISSN 0002-9769. Citado na página 119.
- JUNIOR, Z.; ANTÔNIO, L. *Sistemas de informação baseados na tecnologia web: um estudo sobre seu desenvolvimento*. Tese (Doutorado) — Universidade de São Paulo, 2003. Citado na página 20.
- KAN, S. *Metrics and Models in Software Quality Engineering*. 2nd. ed. [S.l.: s.n.], 2002. Citado na página 46.
- KARELS, M. J. Commercializing open source software. *Queue*, ACM, New York, NY, USA, v. 1, n. 5, p. 40:46–40:55, jul. 2003. ISSN 1542-7730. Disponível em: <<http://doi.acm.org/10.1145/945074.945125>>. Citado na página 36.
- KASIM, N. A.; GUNAWAN, T. Virtual-learning content management system for problem-based learning (PBL) courses. In: *2012 International Conference on Computer and Communication Engineering (ICCCE)*. [S.l.: s.n.], 2012. p. 948–952. Citado na página 120.

- KIATRUANGKRAI, P. et al. Design and Development of Real-Time Communication Content Management System for E-Commerce. In: *2010 Second International Symposium on Data, Privacy and E-Commerce (ISDPE)*. [S.l.: s.n.], 2010. p. 111–116. Citado na página 124.
- KOTONYA, G.; SOMMERVILLE, I. “*Requirements Engineering: Process and Techniques*”. [S.l.: s.n.], 1998. Citado na página 32.
- KUMAR, B. Security in Drupal. In: *2014 3rd International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions)*. [S.l.: s.n.], 2014. p. 1–1. Citado na página 121.
- LARRINAGA, F. et al. A case study on the use of community platforms for inter-enterprise innovation. In: *2011 17th International Conference on Concurrent Enterprising (ICE)*. [S.l.: s.n.], 2011. p. 1–8. Citado na página 123.
- LATIF, N.; HASSAN, M.; HASAN, M. Automated notification and document downloading in E-learning - development of an agent-based framework utilizing the push-pull technology interaction policy. In: *International Symposium on Information Technology, 2008. ITSIm 2008*. [S.l.: s.n.], 2008. v. 1, p. 1–7. Citado na página 123.
- LEITE, J. C. S. P. “*Gerenciando a Qualidade de Software com Base em Requisitos*”. In: *Qualidade de Software: Teoria e Prática*. [S.l.: s.n.], 2001. Citado na página 32.
- LINA, L.; GUOHUA, X. The Construction of Virtual Reference Advisory Platform Based Plone Content Management System. In: *2010 Second International Conference on Computer Engineering and Applications (ICCEA)*. [S.l.: s.n.], 2010. v. 2, p. 529–532. Citado 2 vezes nas páginas 68 e 124.
- LOECHEL, A.; MIHELICIC, G.; PICKL, S. An Open Source Approach for a Military Situational Awareness System. In: *2012 45th Hawaii International Conference on System Science (HICSS)*. [S.l.: s.n.], 2012. p. 1462–1471. Citado 2 vezes nas páginas 68 e 123.
- LONGO, F. et al. Integrating social web and e-learning to enhance cooperation in the building sector. In: *2014 International Conference on Interactive Collaborative Learning (ICL)*. [S.l.: s.n.], 2014. p. 324–329. Citado na página 121.
- MAMANI, N. A. “*Integrando requisitos não funcionais aos requisitos baseados em ações concertas*”. Dissertação (Dissertação de mestrado) — Departamento de informática, PUC-RIO, Maio 1999. Citado 2 vezes nas páginas 11 e 33.
- MARIMUTHU, C.; SEKARAN, K. Software Development for Cloud: An Experiential Study. In: *2013 International Conference on Cloud Computing and Big Data (CloudCom-Asia)*. [S.l.: s.n.], 2013. p. 183–190. Citado 2 vezes nas páginas 68 e 120.
- MARINHO, D. da S.; SOUZA, E. J. de. Pesquisa de qualidade de software brasileiro 2009. *Brasília: SECRETARIA DE POLÍTICA DE INFORMÁTICA, MINISTÉRIO DA CIÊNCIA E TECNOLOGIA*, 2010. Citado na página 46.
- MARKETINGDECONTEÚDO: Blog corporativo para iniciantes: Como escolher um cms. 2014. Disponível em: <<http://marketingdeconteudo.com/como-escolher-um-cms/>>. Acesso em: 19 Ago. 2014. Citado 2 vezes nas páginas 35 e 40.

MAROCO, J.; GARCIA-MARQUES, T. Qual a fiabilidade do alfa de cronbach? questões antigas e soluções modernas? *Laboratório de Psicologia*, v. 4, n. 1, p. 65–90, 2013. Citado 3 vezes nas páginas 90, 91 e 93.

MCGARRY, J. et al. *Practical Software Measurement – Objective Information for Decision Makers*. [S.l.: s.n.], 2002. Citado 3 vezes nas páginas 11, 60 e 63.

MEKPIROON, O. et al. LearnSquare: Thai open-source Learning Management System. In: *5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2008. ECTI-CON 2008*. [S.l.: s.n.], 2008. v. 1, p. 185–188. Citado na página 124.

MELO, M. A. V. de. Requisitos de ferramentas de apoio aos processos de medição de software. S/D. Citado na página 60.

MOLLER, K. H.; PAULISH, D. J. “*Software Metrics, A Practitioners Guide to Improved Product Development*”. [S.l.]: Chapman & Hall, 1993. Citado na página 56.

MORESI, E. *Metodologia da Pesquisa*. 2003. 108 p. Citado na página 24.

NAKWASKI, M.; ZABIEROWSKI, W. Content management system for web portal. In: *2010 International Conference on Modern Problems of Radio Engineering, Telecommunications and Computer Science (TCSET)*. [S.l.: s.n.], 2010. p. 233–235. Citado na página 124.

NATH, M.; ARORA, A. Content management system : Comparative case study. In: *Software Engineering and Service Sciences (ICSESS), 2010 IEEE International Conference on*. [S.l.: s.n.], 2010. p. 624–627. Citado 6 vezes nas páginas 21, 68, 123, 130, 131 e 159.

NBR-ISO/IEC-9126. *NBR ISO/IEC 9126. Engenharia de Software - Qualidade de Produto. Parte 1*. [S.l.]: NBR ISO/IEC, 2003. Citado na página 46.

NIELSEN, J.; LANDAUER, T. K. A mathematical model of the finding of usability problems. *Proceedings ACM/IFIP INTERCHI'93 Conference (Amsterdam, The Netherlands, April 24-29)*, p. 206–213, 1993. Citado na página 160.

NUNNALLY, J. C. *Psychometric theory*. [S.l.]: New York, 1978. Citado na página 91.

NURMINEN, J. et al. Drupal Content Management System on Mobile Phone. In: *5th IEEE Consumer Communications and Networking Conference, 2008. CCNC 2008*. [S.l.: s.n.], 2008. p. 1228–1229. Citado 2 vezes nas páginas 121 e 129.

OPENSOURCECMS: Cms market share. 2014. Disponível em: <<http://www.opensourcecms.com/general/cms-marketshare.php>>. Acesso em: 19 Ago. 2014. Citado 2 vezes nas páginas 35 e 40.

ORDUNA, P. et al. Modelling remote laboratories integrations in e-learning tools through remote laboratories federation protocols. In: *Frontiers in Education Conference (FIE), 2012*. [S.l.: s.n.], 2012. p. 1–6. Citado na página 123.

ORDUNA, P. et al. Generic integration of remote laboratories in learning and content management systems through federation protocols. In: *2013 IEEE Frontiers in Education Conference*. [S.l.: s.n.], 2013. p. 1372–1378. Citado na página 123.

- OWASP. *Cross-Site Request Forgery (CSRF)*. 2015. Disponível em: <https://www.owasp.org/index.php/Cross-Site_Request_Forgery_%28CSRF%29>. Acesso em: 13 May. 2015. Citado na página 128.
- PARK, R.; GOETHERT, W.; FLORAC, W. “*Goal Driven Software Measurement – a Guidebook*”. [S.l.]: Software Engineering Institute, Carnegie Mellon University, 1996. Citado 3 vezes nas páginas 13, 56 e 57.
- PATEL, S.; RATHOD, V.; PARIKH, S. Joomla, drupal and wordpress - a statistical comparison of open source cms. In: *Trendz in Information Sciences and Computing (TISC), 2011 3rd International Conference on*. [S.l.: s.n.], 2011. p. 182–187. Citado 6 vezes nas páginas 13, 43, 44, 45, 119 e 127.
- PATEL, S.; RATHOD, V.; PRAJAPATI, J. Comparative analysis of web security in open source content management system. In: *2013 International Conference on Intelligent Systems and Signal Processing (ISSP)*. [S.l.: s.n.], 2013. p. 344–349. Citado 3 vezes nas páginas 120, 127 e 128.
- PATNAIK, R.; MISHRA, M. Role of Content Management Software (CMS) in libraries for information dissemination. In: *2015 4th International Symposium on Emerging Trends and Technologies in Libraries and Information Services (ETTLIS)*. [S.l.: s.n.], 2015. p. 117–121. Citado 2 vezes nas páginas 120 e 129.
- PERIZZOLO, D. M. Monografia, *Desenvolvimento de um Sistema de Informação Web para Empresas De Factoring*. Blumenau - SC: [s.n.], 2005. 65 f. Citado na página 21.
- POLIC, T.; TOMLJANOVIC, J.; ANIC, N. eAssessment in English for specific purposes. In: *2014 The International Conference on Education Technologies and Computers (ICETC)*. [S.l.: s.n.], 2014. p. 18–23. Citado na página 122.
- PRESSMAN, R. S. “*Software Engineering – A Practitioners Approach*”. Fifth. [S.l.: s.n.], 2001. Citado na página 55.
- PRIYA, T. et al. Mobile interface to content management system based on HTML5 and Drupal: A case study. In: *International Conference on Software Engineering and Mobile Application Modelling and Development (ICSEMA 2012)*. [S.l.: s.n.], 2012. p. 1–6. Citado 2 vezes nas páginas 122 e 130.
- RAMACHANDRAN, R. et al. Talkoot software appliance for collaborative science. In: *Geoscience and Remote Sensing Symposium, 2009 IEEE International, IGARSS 2009*. [S.l.: s.n.], 2009. v. 5, p. V–378–V–381. Citado na página 122.
- REIS, J. C. D. et al. *Uma Prospecção de Plataformas para o Desenvolvimento do Sistema “Todos Nós em Rede”*. [S.l.], 2011. Citado 7 vezes nas páginas 13, 41, 42, 67, 119, 125 e 126.
- SALGADO, F. F.; ATOJI, R. I. Catedral, bazar e software livre. *Catedral, Bazar e Software Livre*, v. 20, 2008. Citado na página 37.
- SANTOS, E. D. Web mining mineração de dados na web utilizando cms. Universidade Jean Piaget de Cabo Verde, 2013. Citado 2 vezes nas páginas 33 e 34.
- SANTOS, R. V. et al. Cope: Correspondencia conceitual entre o modelo psm e a estatística. *sigmaE*, v. 3, n. 1, p. 15–20, 2014. Citado na página 60.

SCHWINGEL, C. A produção de conteúdos no ciberespaço: sistemas de gerenciamento de conteúdos. *SBPJor – Associação Brasileira de Pesquisadores em Jornalismo. VII Encontro Nacional de Pesquisadores em Jornalismo - USP (Universidade de São Paulo)*, Novembro 2009. Citado na página 35.

SERODIO, C. et al. Content server for use in Power Electronics courses. In: *2010 4th IEEE International Conference on E-Learning in Industrial Electronics (ICELIE)*. [S.l.: s.n.], 2010. p. 13–18. Citado 2 vezes nas páginas 68 e 121.

SHERPA: Site. 2014. Disponível em: <<http://www.sherpaglobal.com/>>. Acesso em: 03 Nov. 2014. Citado 2 vezes nas páginas 35 e 36.

SHU, M. et al. Anti-money-laundering system based on mainframe and soa. In: *Computational Intelligence and Communication Networks (CICN), 2013 5th International Conference on*. [S.l.: s.n.], 2013. p. 613–616. Citado na página 19.

SMITH, B. Um guia rápido para gplv3. *Free Software Foundation, Inc. Online: <http://www.gnu.org/licenses/quick-guide-GPLv3.html>*. Patrocinado, v. 4, p. 2008, 2007. Citado na página 41.

SOLLINGEN, R. V.; BERGHOUT, E. *The Goal/Question Metric Method, A Practical Guide For Quality Improvement of Software Development*. [S.l.: s.n.], 1999. Citado 7 vezes nas páginas 11, 13, 55, 58, 59, 87 e 88.

SOMMERVILLE, I. *“Software Engineering”*. fourth edition. [S.l.: s.n.], 1992. Citado 2 vezes nas páginas 11 e 33.

STACCINI, P. et al. A Customized Open Source Content Management System to Support Collaborative Distance Learning: The J@LON Platform. In: *Twentieth IEEE International Symposium on Computer-Based Medical Systems, 2007. CBMS '07*. [S.l.: s.n.], 2007. p. 651–656. Citado na página 123.

SUTHEEBANJARD, P.; PREMCHAIWADI, W. QR-code generator. In: *Knowledge Engineering, 2010 8th International Conference on ICT and*. [S.l.: s.n.], 2010. p. 89–92. Citado na página 122.

SZYBALSKI, M.; ZABIEROWSKI, W. Using web - developer technologies for building an academic service. In: *CAD Systems in Microelectronics (CADSM), 2011 11th International Conference The Experience of Designing and Application of*. [S.l.: s.n.], 2011. p. 301–303. Citado na página 33.

TANENBAUM, A. S.; WOODHULL, A. *Sistemas Operacionais: Projetos e Implementação*. [S.l.]: –, 2006. Citado 2 vezes nas páginas 19 e 20.

TANG, K. et al. NEIMiner: A model driven data mining system for studying environmental impact of nanomaterials. In: *2012 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)*. [S.l.: s.n.], 2012. p. 895–902. Citado na página 120.

TARHAN, A.; YILMAZ, S. G. Systematic analyses and comparison of development performance and product quality of incremental process and agile process. *Elsevier - Information and Software Technology 56 (2014) 477–494*, 2014. Citado na página 54.

TWITTER: Site. 2014. Disponível em: <<https://twitter.com>>. Acesso em: 05 Out. 2014. Citado na página 37.

VASEK, M.; WADLEIGH, J.; MOORE, T. Hacking is not random: a case-control study of webserver-compromise risk. *IEEE Transactions on Dependable and Secure Computing*, PP, n. 99, p. 1–1, 2015. ISSN 1545-5971. Citado 2 vezes nas páginas 122 e 130.

VASQUEZ-RAMIREZ, R.; ALOR-HERNÁNDEZ, G.; POSADA-GOMEZ, R. A hybrid architecture for developing an educational Content Management System. In: *2011 21st International Conference on Electrical Communications and Computers (CONIELECOMP)*. [S.l.: s.n.], 2011. p. 250–255. Citado na página 124.

VISIONWIDGET: - front-end design impression: 30 cms opensource edition. 2014. Disponível em: <<http://visionwidget.com/free-cms-resources.html>>. Acesso em: 02 Set. 2014. Citado na página 35.

WANG, X. et al. Microblogging in Open Source Software Development: The Case of Drupal and Twitter. *IEEE Software*, v. 31, n. 4, p. 72–80, jul. 2014. ISSN 0740-7459. Citado na página 122.

XIANG, C.; YU, W. Using content management system joomla! to build a website for research institute needs. In: *Management and Service Science (MASS), 2010 International Conference on*. [S.l.: s.n.], 2010. p. 1–3. Citado na página 31.

XIANG, C.; YU, W. Using Content Management System Joomla! to Build a Website for Research Institute Needs. In: *2010 International Conference on Management and Service Science (MASS)*. [S.l.: s.n.], 2010. p. 1–3. Citado 2 vezes nas páginas 67 e 120.

YEN, C.-C.; YEN, C.; HSU, J.-S. A Web-based CMS/PDM Integration for Product Design and Manufacturing. In: *IEEE International Conference on e-Business Engineering, 2008. ICEBE '08*. [S.l.: s.n.], 2008. p. 549–553. Citado na página 124.

YOSHIURA, N. Discussion on business of free software products - through e-jikei network project. In: *Instrumentation, Communications, Information Technology, and Biomedical Engineering (ICICI-BME), 2011 2nd International Conference on*. [S.l.: s.n.], 2011. p. 19–22. Citado na página 35.

YUSOFF, M.; JAAFAR, J.; MAHMOOD, A. The visual wiki for the first step of KMS in community college. In: *National Postgraduate Conference (NPC), 2011*. [S.l.: s.n.], 2011. p. 1–6. Citado na página 124.

Apêndices

APÊNDICE A – Detalhamento das Expressões de Busca elaboradas

Neste apêndice é apresentado o detalhamento das expressões de busca elaboradas para o levantamento de características apresentado no apêndice B.

O raciocínio usado para a construção das Strings de pesquisa é apresentado a seguir :

1. Definir uma subquestão de pesquisa:

Quais seriam as características relevantes na escolha de um determinado CMS levando em consideração um contexto específico?

2. Separar palavras chaves:

CMS , características, atributos

3. Elaborar a *String* de Busca com base nas palavras chaves:

(“Content manager system “ OR “CMS” OR “Sistema gerenciador de conteúdo”) AND (“features” OR “attributes” OR “appearance”)

4. Submeter a *String* de Busca em alguma base de dados conhecida:

- IEEE;
- ACM;
- Science Direct;
- Springer
- Scopus

5. Refinar a String de Pesquisa caso a quantidade de artigos retornados seja superior a 60.

Para o teste das Expressões de Busca foi escolhida a base de dados IEEE, pois a IEEE oferece inúmeros filtros que facilitam o refinamento da Expressão de Busca, conforme a necessidade.

A.1 Resultados para os CMS mais populares

Após a aplicação da *String* de Pesquisa na base de dados IEEE foram obtidos os seguintes resultados:

(“Content manager system “ OR “CMS” OR “Sistema gerenciador de conteúdo”) AND (“features” OR “attributes” OR “appearance”) foram retornados 545.871 resultados.

Para refinar a *String* de pesquisa foram adicionados os CMSs mais populares gerando uma nova *String* de Pesquisa

(“Content manager system“ OR “CMS” OR “Sistema gerenciador de conteúdo”) AND (“features” OR “attributes” OR “appearance”) AND (“WordPress” OR “Joomla” OR “Drupal”)

Aplicando-se a *String* de pesquisa novamente na base de dados IEEE foram obtidos 999 resultados.

Dentre esses resultados foi observado que muitos artigos encontrados não estavam relacionados com CMS, nem se quer com computação, sendo assim a *String* de pesquisa foi ajustada para:

(“Content AND manager AND system “ OR “CMS” OR “Sistema gerenciador de conteúdo”) AND (“features” OR “attributes” OR “appearance”) resultando em 58 resultados.

Após a aplicação dos critérios de exclusão da seção X restaram 32 artigos.

Porém para o levantamento de características não bastou apenas considerar somente os CMSs mais populares, foi necessário confrontar características de CMSs não populares para o estudo. Então foi criada uma nova *string* de pesquisa para filtrar artigos que tivessem o foco de CMSs não populares.

A.2 Resultados para os CMS não populares

Para esta etapa foi feita uma nova *String* de pesquisa que exclui os CMS populares e acrescenta a característica *open source*.

("CMS"OR "Content Management System") NOT “WordPress” NOT “Joomla” NOT “Drupal” AND ("features"AND "open source")

Após a aplicação constatou-se 715 resultados considerando todo o texto, abstract e título. Porém o resultado não foi satisfatório e o filtro de apenas abstract e título do *Advanced Search* do IEEE foi ativado mudando a quantidade de resultados para apenas 8. Um resultado considerado satisfatório.

Nessa fase os critérios de seleção apresentados na seção A.3 foram mantidos e dois artigos dos 8 selecionados foram descartados.

Porém mais um teste foi feito trocando-se a palavra *features* na String de pesquisa por *appearance* a fim de levantar mais artigos para a pesquisa e a nova String:

```
((("CMS"OR "Content Management System") NOT "Wordpress" NOT "Joomla" NOT "Drupal" AND ("appearance"AND "open source")))
```

 retornou 97 resultados.

O que não foi satisfatório. Como solução foi aplicado o filtro *Publication Year* do próprio IEEE considerando apenas artigos dos anos de 2010 a 2014 e assim a quantidade de arquivos retornados caiu para 49.

Dos 49 artigos investigados apenas 6 artigos foram selecionados com a aplicação dos critérios de exclusão da seção A.3.

A.3 Critérios de Exclusão para os artigos selecionados

A partir dos 58 artigos selecionados foram definidos critérios de exclusão para uma melhor seleção de características, sendo assim foram excluídos:

- Artigos que não mencionavam em seu título ou abstract a palavra CMS, ou *Content Management System*, ou Drupal, ou Wordpress ou Joomla, ou demais palavras relacionadas.
- Artigos fechados.
- Não apresentar, ou não deixar claro no texto a existência de pelo menos 5 características de CMSs.

APÊNDICE B – Levantamento de Características de CMS - Tabelas

B.1 Contexto de artigos Lidos

B.1.1 Contexto para artigos envolvendo CMSs mais populares

Tabela 21 – Contexto para artigos levantados.

Id	Título	Contexto
1	Comparativo de Ferramentas de Criação e Gestão de Web Sites CMS (Content Management System).	Desenvolvimento do Web Site do Tribunal de Contas do Estado do Pará. (COSTA; CAMINHA; LIMA, 2012)
2	WordPress as Library CMS.	Criação de uma biblioteca virtual usando WordPress.(JONES; FARRINGTON, 2011)
3	Uma Prospecção de Plataformas para o Desenvolvimento do Sistema “Todos Nós em Rede”.	Construção de uma rede social para professores com o objetivo de apoiar a comunicação e a práticas profissionais do dia a dia. (REIS et al., 2011)
4	Building Computer System for Musical School with CSM Plone.	Construção de um sistema para uma escola de música usando o CMS Plone.(GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)
5	Joomla. Drupal and WordPress - A Statistical Comparison of Open Source CMS.	Testar o desempenho de 3 CMS escolhidos: Caso 1: montagem de página com adição de plugins. Caso 2 : mesmo caso 1 só que com ambiente diferenciado.(PATEL; RATHOD; PARIKH, 2011)

6	Comparative analysis of web security in open source content management system.	Testar a segurança de 3 CMS escolhidos: Caso 1- Criar páginas comuns e submeter tais páginas a ataques WEB ex: SQL injection. No caso 2 foi utilizado Acunetix WVS Reporter v6.0 para descobrir o potencial de segurança em diferentes CMS.(PATEL; RATHOD; PRAJAPATI, 2013)
7	Using Content Management System Joomla! to Build a Website for Research Institute Needs.	Alteração de Templates no Joomla!.(XIANG; YU, 2010b)
8	Optimization model of management and security of multimedia content using Drupal CMS.	Uso do Drupal para a construção de sites com segurança para multimídia (ENNERT; CAJKOVSKY; ADAM, 2012)
9	The Web Development Based on the Drupal System.	Apresentação de características do drupal e desenvolvimento de um website em seguida.(CHENG; WANG, 2012)
10	Role of Content Management Software (CMS) in libraries for information dissemination.	Utilização do CMS Joomla para a construção de uma biblioteca virtual.(PATNAIK; MISHRA, 2015)
11	Software Development for Cloud: An Experiential Study.	Desenvolvimento em nuvem usando CMS.(MARIMUTHU; SEKARAN, 2013)
12	Virtual-learning content management system for problem-based learning (PBL) courses.	Sistema baseado em CMS para o ensino de PBL.(KASIM; GUNAWAN, 2012)
13	NEIMiner: A model driven data mining system for studying environmental impact of nanomaterials.	Sistema de mineração de dados usando o CMS Drupal.(TANG et al., 2012)
14	A game with a purpose for annotating Greek folk music in a web content management system.	Sistema para hospedagem de música e protótipo de jogo.(GIOUVANAKIS et al., 2013)
15	Designing of e-commerce system by CMS Joomla software.	Utilização do Joomla para a construção de um site para e-commerce.(DOROSH; KUCHMIJ, 2009)

16	Realization of electronic textbook by means of Drupal Content Management System.	Uso do CMS Drupal para a produção de livros eletrônicos.(ILIEV; VASILEVA, 2013)
17	The development of a new branch of the economy: e-commerce. Practical use with Content Management Systems Joomla! and enlargement VirtueMart.	Uso do Joomla para a construção de um site de e-commerce.(HETKA; ZABIEROWSKI; NAPIERALSKI, 2009)
18	Exploring possibilities to analyse microblogs for dependability information in variability-intensive open source software systems.	Utilização de microblogs para análise de confiabilidade de sistemas de software livre.(GALSTER; TOFAN, 2013)
19	Extending web content management systems navigation capabilities with semantic navigation maps.	Técnicas de recuperação de informação em CMS.(DISTANTE; RISI; SCANNIELLO, 2010)
20	Integrating social web and e-learning to enhance cooperation in the building sector.	Criação de uma Plataforma de Aprendizagem Colaborativa.(LONGO et al., 2014)
21	Security in Drupal.	Segurança no Drupal. (KUMAR, 2014)
22	Integration of captive and JOOMLA: Online demonstration and assessment eLearning resources.	Integração do Joomla com o Adobe captive em uma plataforma web para ensino multimídia.(AW et al., 2010)
23	A Multi-sites Scheme Based on Open Source CMS, Drupal.	Uso de Drupal para construção de sistemas multi-Loais.(JINWEI; JIANHONG; HAICHENG, 2010)
24	Content server for use in Power Electronics courses.	Implementação de um portal web para auxílio de estudantes. (SERODIO et al., 2010)
25	Drupal Content Management System on Mobile Phone.	Construindo um site com auxílio do drupal para tecnologia mobile.(NURMINEN et al., 2008)
26	An API for Quran portal using Drupal technology.	Uso do Drupal para desenvolvimento de aplicação em nuvem. (ADHONI; HAMAD, 2014)

27	Learning 2.0 in the Information Systems Curriculum.	Criação de um sistema para aprendizagem com o auxílio do Drupal.(ALBRECHT; MATTHEWS, 2009)
28	Supporting Agricultural Communities with Workflows on Heterogeneous Computing Resources.	Solução web para comunidades agrícolas envolvendo grand e processamento de dados(BALASKO et al., 2014)
29	Automatically classifying and interpreting polar datasets with Apache Tika.	Extração de meta dados em sistema para a análise de ambiente na antartida.(BURGESS; MATTMANN, 2014)
30	Advanced solution for delivering educational multimedia content based on content management system.	Aplicação multimidia usando o Drupal.(CYMBALAK; JAKAB; MICHALKO, 2012)
31	My-Plant.org: A phylogenetically structured social network.	Criação de uma rede social com auxílio de Drupal.(HANLON et al., 2010)
32	eAssessment in English for specific purposes.	Estudo de caso para construir de um sistema de avaliação para alunos de nível superior. .(POLIC; TOMLJANOVIC; ANIC, 2014)
33	Mobile interface to content management system based on HTML5 and Drupal: A case study.	Fornecer conteúdo multimídia para dispositivos móveis.(PRIYA et al., 2012)
34	Talkoot software appliance for collaborative science.	Construção de Rede Social.(RAMACHANDRAN et al., 2009)
35	A Semantic Web Approach in the Implementation of a Linked Data Portal Using a CMS.	Construção de um framework envolvendo web semantica.(GIANNOPOULOU et al., 2014)
36	Hacking is not random: a case-control study of webserver-compromise risk.	Fatores de risco do uso de cms em web services. (VASEK; WADLEIGH; MOORE, 2015)
37	Microblogging in Open Source Software Development: The Case of Drupal and Twitter.	Twwitter e Drupal.(WANG et al., 2014)
38	QR-code generator.	Gerador de QR Codes.(SUTHEEBANJARD; PREM-CHAIWADI, 2010)

39	A case study on the use of community platforms for inter-enterprise innovation.	Projeto de pesquisa para a implementação de um sistema para um ambiente corporativo.(LARRINAGA et al., 2011)
40	Modelling remote laboratories integrations in e-learning tools through remote laboratories federation protocols.	Laboratorios remotos educativos.(ORDUNA et al., 2012)
41	Generic integration of remote laboratories in learning and content management systems through federation protocols.	Laboratorios remotos educativos.(ORDUNA et al., 2013)

B.1.2 Contexto para artigos envolvendo CMSs não populares

Tabela 22 – Contexto para artigos levantados CMSs não populares.

Id	Título	Contexto
1	Content management system : Comparative case study.	Estudo de Características em CMSs Open source. (NATH; ARORA, 2010)
2	A Customized Open Source Content Management System to Support Collaborative Distance Learning: The J@LON Platform.	Sistema para ensino a distancia com auxilio de zope plone.(STACCINI et al., 2007)
3	An Open Source Approach for a Military Situational Awareness System An Open Source Approach for a Military Situational Awareness System.	Discussão de falhas em sistema militares. Sistemas que envolvem o uso de CMS. (LOECHEL; MIHELICIC; PICKL, 2012)
4	Automated notification and document downloading in E-learning - development of an agent-based framework utilizing the push-pull technology interaction policy.	Sistema para E-learning utilizando CMS. (LATIF; HASSAN; HASAN, 2008)
5	Construction of education application platform based on Plone content management system.	Construção de uma aplicação educacional com o auxílio de zope plone.(JIUGEN; RUONAN; QI, 2012)

6	An Open Source Approach for a Military Situational Awareness System An Open Source Approach for a Military Situational Awareness System.	Construção de sistema para e-commerce. (KIATRUANG-KRAI et al., 2010)
7	Content management system for web portal.	Informações básicas sobre CMS para um portal web.(NAKWASKI; ZABIEROWSKI, 2010)
8	The Construction of Virtual Reference Advisory Platform Based Plone Content Management System.	Utilização do plone para construção de sistema de aconselhamento virtual.(LINA; GUOHUA, 2010)
9	A Web-based CMS/PDM Integration for Product Design and Manufacturing.	Aplicação prática para integrar CMS e PDM (Product Data Management). (YEN; YEN; HSU, 2008)
10	LearnSquare: Thai open-source Learning Management System.	Ilustrar as características do sistema LearnSquare.(MEKPIROON et al., 2008)
11	A hybrid architecture for developing an educational Content Management System.	Reutilização e gerenciamento de conteúdo educacional usando CMS e LMS.(VASQUEZ-RAMIREZ; ALOR-HERNÁNDEZ; POSADA-GOMEZ, 2011)
12	The visual wiki for the first step of KMS in community college.	Explorando fatores de sucesso em CMSs no contexto de sistemas para aprendizagem. (YUSOFF; JAAFAR; MAHMOOD, 2011)

B.2 Características de CMSs mais populares

Tabela 23 – Características de CMSs levantadas para CMSs populares.

Id	Característica	Descrição
1	Facilidade de instalação	Ato de transferir os arquivos baixados para o sistema operacional, realizar as devidas configurações e executar o programa instalado em uma maquina local. (COSTA; CAMINHA; LIMA, 2012)
2	Usabilidade	Facilidade com que as pessoas podem empregar uma ferramenta ou objeto a fim de realizar uma tarefa específica e importante. (COSTA; CAMINHA; LIMA, 2012)
3	Flexibilidade	Ter os recursos necessários para evoluir de acordo com suas funcionalidades. (REIS et al., 2011)
4	Segurança da ferramenta	Gerenciamento de segurança, incluindo gerenciamento de sessão e controle de acesso baseado em função de conteúdos / operações. (DISTANTE; RISI; SCAN-NIELLO, 2010)
5	Interatividade	Processo bem organizado de criação de conteúdos.(ILIEV; VASILEVA, 2013)
6	Suporte a serviços mobiles(Smathphones, tablets)	Permitir o uso do CMS em dispositivos mobiles como smarthphones e tablets (JONES, 2008)
7	Documentação	Apresentar guia de instalação, guia de início rápido, tutoriais, wikis, fóruns de discussão de desenvolvedores, entre outros. (REIS et al., 2011)
8	Conformidade com os padrões World Wide Web Consortium	Conformidade com padrões para a criação e interpretação de conteúdos para WeB (REIS et al., 2011)
9	Taxonomia	Forma com que os CMS organizam seu conteúdo. O Joomla por exemplo define a organização de seu conteúdo em "seções"e "categorias". (COSTA; CAMINHA; LIMA, 2012)

10	Liberdade para alteração de um template específico	O usuário tem liberdade para editar, ou acrescentar conteúdo em um determinado template. (REIS et al., 2011)
11	Configuração para interagir com outros sites	Possuir a capacidade para se comunicar com outras páginas web. (REIS et al., 2011)
12	Edição de conteúdo em qualquer aplicativo (como o Dreamweaver ou Word) sem ter que mover o conteúdo do servidor para o desktop	O conteúdo pode ser editado diretamente em qualquer outro software (Word, Dreamweaver), sem a necessidade de download para o desktop (GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)
13	Existência de recursos para controle de acesso	Controlar o acesso a informações individuais e dados com base em papéis definidos, grupos de usuários. (Definição de quais usuários ou grupos de usuários podem visualizar, editar, publicar dados individuais, etc (ENNERT; CAJKOVSKY; ADAM, 2012)
14	Permitir upload de uma quantidade grande de tipos de arquivos (Word, Excel, RTF, PDF)	Disponibilização de vários tipos de arquivos no conteúdo da página. (GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)
15	Permitir o gerenciamento de conteúdo de qualquer navegador(Até os de celular)	Acesso ao CMS por qualquer browser. Até browsers de dispositivos móveis como tablets e smartphones (GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)
16	Facilidade em criar novos conteúdos	A criação de conteúdos deve ocorrer sem qualquer habilidade com linguagens de programação. (GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)
17	Boa capacidade para realizar pesquisas, buscas	Suporte para uso simples, armazenamento, pesquisa e dados recuperação por critérios diferentes e filtros. (ENNERT; CAJKOVSKY; ADAM, 2012)
18	Existência de Templates e Plugins disponíveis para a alteração do site em questão	O CMS deve dispor de temas para a modificação do seu Layout e plugins para a adição de funcionalidades.(GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)

19	Os dados podem ser acessados ou armazenados em bancos de dados relacionais	Os dados podem ser armazenados, ou acessados a partir de bancos de dados relacionais. Exemplos: Oracle, SQLServer, PostgreSQL, MySQL, Interbase, e bancos de dados compatíveis com ODBC. (GAJDKA; ZABIEROWSKI; NAPIERALSKI, 2008)
20	Tempo de carregamento da página	Em ms (milisegundos)(PATEL; RATHOD; PARIKH, 2011)
21	Tamanho da Página	Tamanho total da página em KB. (PATEL; RATHOD; PARIKH, 2011)
22	Total de Requisições	Número de requisições enviadas ao servidor ao se carregar uma página. (PATEL; RATHOD; PARIKH, 2011)
23	Total de arquivos JS	Número de arquivos <i>Java Script</i> usados pelo CMS para fazer uma página. (PATEL; RATHOD; PARIKH, 2011)
24	Total de arquivos CSS	Número de arquivos CSS usados pelo CMS para fazer uma página. (PATEL; RATHOD; PARIKH, 2011)
25	Tempo em que se armazena o conteúdo na memória cache antes da página carregar	Chamado de PLT.(PATEL; RATHOD; PARIKH, 2011)
26	Quantidade de dados armazenados na memória cache antes da página carregar	Chamado de PS (PATEL; RATHOD; PARIKH, 2011)
27	Existência de recursos de proteção a ataques do tipo SQL Injection	Tipo de ataque à segurança web no qual o atacante adiciona código SQL através da caixa de entrada de formulário web para ganhar acesso aos dados e realizar mudanças. (PATEL; RATHOD; PRAJAPATI, 2013)
28	Existência de recursos de proteção a ataques do tipo XXS	Forma de ataque por meio de conteúdos dinâmicos (gifs, banners, entre outros.) que disparam scripts maliciosos, roubam <i>cookies</i> , ou enviam solicitações de conteúdo não autorizado.(PATEL; RATHOD; PRAJAPATI, 2013)

29	Existência de recursos de proteção a ataques do tipo Remote File Inclusion RFI	Com auxílio de um arquivo remoto normalmente chamado de SHELL, o <i>hacker</i> pode obter direitos de administrador do servidor. A shell é um arquivo interface gráfica do usuário que é usado para visitar os arquivos remotos e executar seu próprio código em servidores da web. (PATEL; RATHOD; PRAJAPATI, 2013)
30	Existência de recursos de proteção a ataques do tipo LFI	<i>Local File Inclusion</i> é uma forma de ataque que consiste em incluir arquivos em um servidor através do navegador web. (PATEL; RATHOD; PRAJAPATI, 2013))
31	Existência de recursos de proteção a ataques do tipo Directory Transversal	Método de invasão no qual os <i>hackers</i> se baseiam em ataques de passagem de diretório para tentar o acesso a arquivos do servidor Web de acesso restrito e que estão fora do diretório raiz. (PATEL; RATHOD; PRAJAPATI, 2013)
32	Existência de recursos de proteção a ataques do tipo <i>Cookie Poisoning</i>	Ataques de <i>Cookie Poisoning</i> envolvem a modificação do conteúdo de um <i>cookie</i> (informações pessoais armazenadas no computador de um usuário da Web), a fim de contornar os mecanismos de segurança. Usando ataques de envenenamento de cookie, os atacantes podem obter informações não autorizadas sobre outro usuário e roubar sua identidade (IMPERVA, 2015).
33	Existência de recursos de proteção a ataques do tipo <i>Cross-Site Request Forgery (CSRF)</i>	É um ataque que força um usuário autenticado em uma página web a executar ações indesejadas (OWASP, 2015)
34	Ser software Livre ou não	O CMS é open source ou proprietário. Preocupação com o licenciamento do CMS. (CHENG; WANG, 2012)
35	Popularidade	Popularidade do CMS. O conceito popularidade é explicado na seção 3.4.2
36	Taxa de atualização	Frequência com que o CMS oferece atualizações incorporando novos recursos e sistemas de apoio. (ILIEV; VASILEVA, 2013)
37	Tamanho da Comunidade	Apoio da comunidade para a solução de problemas. Desenvolvedores contribuem para a evolução e manutenção do CMS. (CHENG; WANG, 2012)

38	Simplicidade de acesso	Os dados em um CMS podem ser definidos de várias formas, sem muitas restrições, facilitando o acesso. (ENNERT; CAJKOVSKY; ADAM, 2012)
39	Existência de recursos para vários idiomas	Suporte a vários idiomas distintos. (ENNERT; CAJKOVSKY; ADAM, 2012)
40	Existência de recursos para gerência de versões de conteúdo	Controle de versão, atualizações e revisões. (ENNERT; CAJKOVSKY; ADAM, 2012)
41	Permitir suporte para conteúdos multimídia	Visualizar, partilhar, catalogar e permitir gestão de conteúdo multimídia (vídeos, fotos, músicas)(ENNERT; CAJKOVSKY; ADAM, 2012)
42	Modularidade	Quebra de funcionalidades por meio de várias extensões diferentes. (ILIEV; VASILEVA, 2013)
43	Permitir interação com várias ferramentas	O CMS pode ser usado em conjunto com diversas ferramentas. Exemplos: Frameworks, sistemas operacionais, bancos de dados, entre outros. (NURMINEN et al., 2008)
44	Custo	Redução de custos com o desenvolvimento local e com manutenção. (ILIEV; VASILEVA, 2013)
45	Manutenção	Melhorias na manutenção evitando duplicação de conteúdo. (ILIEV; VASILEVA, 2013)
46	Ser Multiplataforma	Capacidade de se adaptar a diversos tipos de sistemas. Exemplos: mobiles, sistemas operacionais. (NURMINEN et al., 2008)
47	Fácil Utilização	Sem conhecimento de programação web é possível criar um site em um curto período de tempo. (PATNAIK; MISHRA, 2015)
48	Não requer conhecimento de Linguagem de Programação para uso	Não é necessário o conhecimento de linguagens de programação para a criação de um web site. (DISTANTE; RISI; SCANNIELLO, 2010)
49	Variabilidade	Grande quantidade de módulos, plugins, templates e outros recursos. (GALSTER; TOFAN, 2013)

50	Confiabilidade	Probabilidade da operação de um software estar livre de falhas por um período determinado de tempo em um ambiente específico. (GALSTER; TOFAN, 2013)
51	Modificabilidade	Capacidade de prover recursos para modificar o site em questão conforme a necessidade do usuário. (CYMBALAK; JAKAB; MICHALKO, 2012)
52	Extensibilidade	O CMS pode ser estendido por meio de seus módulos. (ENNERT; CAJKOVSKY; ADAM, 2012)
53	Ciclo de desenvolvimento rápido (Rad)	Adaptação ao Ciclo de Desenvolvimento Rápido (RAD). (HANLON et al., 2010)
54	Arquitetura flexível e extensível	A arquitetura é modularizada e facilmente extensível. (HANLON et al., 2010)
55	Capacidade de Reuso	Módulos do CMS podem ser reutilizados conforme a necessidade do desenvolvedor. (PRIYA et al., 2012)
56	Mecanismo anti phishing	Mecanismo contra fraudes, na qual o fraudador tenta adquirir dados pessoais, como senhas, dados financeiros, por meio de SMS, emails entre outros. (VASEK; WADLEIGH; MOORE, 2015)
57	Violação de segurança por redirecionamento de página	Uma quebra de segurança pode ocorrer quando ao clicar em um determinado link ocorre um direcionamento de página. (VASEK; WADLEIGH; MOORE, 2015)

B.3 Características de CMSs não populares

Tabela 24 – Características de CMSs levantadas para CMSs não populares.

Id	Característica	Descrição
1	Geração automática de Interface de Usuário	Interface com o usuário é gerada automaticamente. (NATH; ARORA, 2010)

2	Inscrição automática de usuário	Permite que o usuário entre automaticamente no momento da abertura do site.(NATH; ARORA, 2010)
3	Controle dinâmico	Permitir a manipulação dos dados fornecidos (NATH; ARORA, 2010)
4	Notificação de senha esquecida	Este recurso suporta usuários que esqueceram sua senha. O CMS notifica o usuário com uma nova senha aleatória (NATH; ARORA, 2010)
5	Confirmação por email	Confirma uma nova identificação de usuário (ID) para o usuário (NATH; ARORA, 2010)
6	Backup	Facilidade de backup de dados para um usuário salvar seus dados.(NATH; ARORA, 2010)
7	Gerenciamento de usuários	Gerencia as informações da dos usuários e suas respectivas autenticações.(NATH; ARORA, 2010)
8	Geração automática de código	Código da página é gerado automaticamente. (NATH; ARORA, 2010)
9	Permitir múltiplos usuários	Múltiplo gerenciamento de acesso do usuário por meio de de travamento.(NATH; ARORA, 2010)
10	Permitir download de arquivos	Gestão dos downloads de arquivos por um usuário.(NATH; ARORA, 2010)
11	Permitir gerenciamento de arquivos	Gestão dos arquivos enviados pelo usuário.(NATH; ARORA, 2010)
12	Comentários	Facilidade para fornecer comentários.(NATH; ARORA, 2010)
13	Mecanismo anti plágio	Permite que um site identifique se existe conteúdo plagiado em seu sistema.(NATH; ARORA, 2010)
14	Fóruns	Suporte a fóruns de discussão.(NATH; ARORA, 2010)
15	Verificação de Formato	Recurso que pode ser usado onde o conteúdo pode ser entregue em dois ou mais formatos como qualquer documento pode ser entregue em pdf ,. doc ,. tex etc. ou arquivos de mídia podem ser entregues em. avi ,. dat, etc identificação automática do formato e submetido a conversão para o formato desejado.(NATH; ARORA, 2010)

B.4 Cruzamento de Características de CMS com características SQuaRE

A tabela 25 apresenta as características de CMS levantadas com as características relacionadas a qualidade do produto e as suas respectivas subcaracterísticas. Já a tabela 26 apresenta o mesmo raciocínio só que sob a perspectiva de qualidade em uso.

Tabela 25 – Cruzamento de características de CMS com características SQuaRE - Qualidade do Produto.

Característica CMS	Qtd	Característica SQuaRE	Subcaracterística SQuaRE
Ser software Livre ou não	40	Não se aplica a SQuaRE	Não se aplica a SQuaRE
Facilidade para interação com várias ferramentas	38	Adequação Funcional	Completude Funcional
Modularidade	32	Manutenibilidade	Modularidade
Os dados podem ser acessados ou armazenados em bancos de dados relacionais	28	Adequação Funcional	Completude Funcional
Existência de recursos para controle de acesso	25	Segurança	Autenticidade
Existência de Templates e Plugins disponíveis para a alteração do site em questão	19	Não se aplica a SQuaRE	Não se aplica a SQuaRE
Popularidade	17	Não se aplica a SQuaRE	Não se aplica a SQuaRE
Boa capacidade para realizar pesquisas, buscas	17	Adequação Funcional	Completude Funcional
Configuração para interagir com outros sites	16	Adequação Funcional	Completude Funcional
Facilidade em criar novos conteúdos	16	Usabilidade	Operacionalidade
Interatividade	15	Usabilidade	Operacionalidade
Usabilidade	14	Usabilidade	Característica Square
Segurança da Ferramenta	14	Segurança	Característica Square
Tamanho da Comunidade	13	Não se aplica a SQuaRE	Não se aplica a SQuaRE

Liberdade para alteração de um template específico	13	Manutenibilidade	Modificabilidade
Permitir suporte para conteúdos multimídia	13	Adequação Funcional	Compleitude Funcional
Ser multiplataforma	13	Portabilidade	Adaptabilidade
Permitir o gerenciamento de conteúdo de qualquer navegador(Até os de celular)	12	Portabilidade	Adaptabilidade
Manutenção	11	Manutenibilidade	Característica SQuaRE
Permitir upload de uma quantidade grande de tipos de arquivos (Word, Excel, RTF, PDF)	10	Adequação Funcional	Compleitude Funcional
Existência de recursos para a gerência de versão de conteúdo	10	Adequação Funcional	Compleitude Funcional
Modificabilidade	10	Manutenibilidade	Modificabilidade
Arquitetura flexível e extensível	9	Não se aplica a SQuaRE	Não se aplica a SQuaRE
Fácil Utilização	9	Usabilidade	Operacionalidade
Gerenciamento de usuários	9	Adequação Funcional	Compleitude Funcional
Permitir download de arquivos	8	Adequação Funcional	Compleitude Funcional
Permitir gerenciamento de arquivos	8	Adequação Funcional	Compleitude Funcional
Taxonomia	7	Manutenibilidade	Modificabilidade
Facilidade de instalação	7	Portabilidade	Instalabilidade
Permitir múltiplos usuários	6	Adequação Funcional	Compleitude Funcional
Não requer conhecimento de Linguagem de Programação para uso	6	Usabilidade	Aprendizado
Geração automática de Interface de Usuário	5	Adequação Funcional	Compleitude Funcional
Edição de conteúdo em qualquer aplicativo (como o Dreamweaver ou Word) sem ter que mover o conteúdo do servidor para o desktop	5	Adequação Funcional	Compleitude Funcional
Existência de recursos de proteção a ataques do tipo SQL Injection	4	Segurança	Integridade
Variabilidade	4	Adequação Funcional	Compleitude Funcional
Extensibilidade	4	Adequação Funcional	Compleitude Funcional

Backup	4	Adequação Funcional	Completude Funcional
Documentação	3	Usabilidade	Aprendizado
Tempo em que se armazena o conteúdo na memória cache antes da página carregar	3	Performance	Comportamento do Tempo
Quantidade de dados armazenados na memória cache antes da página carregar	3	Performance	Utilização de Recursos
Fóruns	3	Adequação Funcional	Completude Funcional
Suporte a serviços mobiles (Smathphones tablets...)	2	Portabilidade	Adaptabilidade
Tempo de carregamento da página	2	Eficiência de Desempenho	Comportamento do tempo
Total de Requisições	2	Eficiência de Desempenho	Utilização de Recursos
Existência de recursos de proteção a ataques do tipo Proteção contra XXS	2	Segurança	Integridade
Existência de recursos de proteção a ataques do tipo Cookie Poisoning	2	Segurança	Integridade
Existência de recursos de proteção a ataques do tipo Cross-Site Request Forgery (CSRF)	2	Segurança	Integridade
Simplicidade de Acesso	2	Usabilidade	Operacionalidade
Uso simples de várias opções de idioma	2	Adequação Funcional	Completude Funcional
Geração automática de código	2	Adequação Funcional	Completude Funcional
Inscrição automática de usuário	2	Adequação Funcional	Completude Funcional
Controle dinâmico (manipulação de dados fornecidos)	2	Adequação Funcional	Completude Funcional
Comentários	2	Adequação Funcional	Completude Funcional
Tamanho da Página	2	Eficiência de Desempenho	Utilização de Recursos
Conformidade com os padrões World Wide Web Consortium	1	Usabilidade	Acessibilidade
Total de arquivos JS	1	Eficiência de Desempenho	Utilização de Recursos

Total de arquivos CSS	1	Eficiência de Desempenho	Utilização de Recursos
Existência de recursos de proteção a ataques do tipo RFI (Remote File Inclusion)	1	Segurança	Integridade
Existência de recursos de proteção a ataques do tipo LFI	1	Segurança	Integridade
Existência de recursos de proteção a ataques do tipo Directory Traversal	1	Segurança	Integridade
Confiabilidade	1	Confiabilidade	Característica Square
Ciclo de desenvolvimento rápido (Rad)	1	Não se aplica a SQuaRE	Não se aplica a SQuaRE
Capacidade de Reuso	1	Manutenibilidade	Reusabilidade
Mecanismo anti phishing	1	Segurança	Integridade
Violação de segurança por redirecionamento de página	1	Segurança	Integridade
Notificação de senha esquecida	1	Adequação Funcional	Compleitude Funcional
Geração automática de código	1	Adequação Funcional	Compleitude Funcional
Mecanismo anti plágio	1	Segurança	Integridade
Verificação de Formato	1	Adequação Funcional	Compleitude Funcional
Confirmação por email	1	Adequação Funcional	Compleitude Funcional

Tabela 26 – Cruzamento de características de CMS com características SQuaRE - Qualidade em Uso

Característica CMS	Qtd	Característica SQuaRE	Subcaracterística SQuaRE
Flexibilidade	18	Cobertura de Contexto	Flexibilidade
Custo	10	Inexistência de riscos	Mitigação de riscos economicos

B.5 Rastreabilidade de Características - Figuras

C/A	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41																
31					x																																																				
32					x																																																				
33					x																																																				
34	x	x						x	x	x	x	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		x	x	x	x	x																		
35							x	x	x			x	x			x	x	x						x	x	x	x	x	x						x	x			x	x																	
36								x								x																																									
37							x	x	x			x							x									x			x																										
38								x																																																	
39								x												x																																					
40								x								x				x																																					
41								x		x						x							x																																		
42							x	x	x	x	x			x	x	x	x	x	x																																						
43								x	x	x	x	x	x		x				x	x	x																																				
44									x																																																
45									x																																																
46									x	x																																															
47									x	x																																															
48													x																																												
49																																																									
50																																																									
51																																																									
52																																																									
53																																																									
54																																																									
55																																																									
56																																																									
57																																																									

Figura 55 – Rastreabilidade de Características para CMSs populares Parte 2

C/A	Citada por											
	1	2	3	4	5	6	7	8	9	10	11	12
1					x			x				
2	x		x			x		x				
3			x					x	x			
4			x		x		x			x		
5		x			x							
6			x									
10			x					x				
11			x									x
12			x		x							
13	x	x	x		x		x	x				x
14	x	x	x	x	x					x	x	x
15	x		x									
16					x	x		x				
17			x					x				x
18	x	x	x					x				
19	x	x	x	x		x	x		x	x	x	
34		x	x	x	x	x	x	x	x	x		x
35									x			
37			x									
39								x				
40	x		x			x					x	x
41	x	x									x	x
42			x		x		x	x	x	x		
43	x	x	x	x		x	x	x	x	x	x	
44	x		x			x	x					x
45	x	x	x									x
46	x				x			x				
47	x	x	x					x		x		
48					x	x						x
51			x			x	x	x		x		
52												x
53												x
54			x			x	x				x	

Figura 56 – Rastreabilidade de Características para CMSs não populares Parte 1

C/A	Citada por											
	1	2	3	4	5	6	7	8	9	10	11	12
1	x		x		x	x				x		
2	x		x									
3	x		x									
4	x											
5	x											
6	x		x							x		x
7	x	x	x		x		x	x	x	x	x	x
8	x											
9	x						x	x		x	x	x
10	x	x		x	x			x		x	x	x
11	x	x	x	x	x			x			x	x
12	x									x		
13	x											
14	x									x		x
15				x								

Figura 57 – Rastreabilidade de Características para CMSs não populares Parte 2

A figura 57 mostra somente o mapeamento para as novas características.

APÊNDICE C – Questionários

C.1 Questionário 1 - Levantamento de Características de CMSs

Este formulário faz parte de um trabalho de conclusão de curso, cujo objetivo é identificar características importantes de Sistemas gerenciadores de conteúdo para servirem de apoio a quem vai escolher um determinado produto de CMS para ser usado no desenvolvimento de um site WEB. Ele se baseia em características pesquisadas na literatura como sendo importantes para produtos de CMS. Ao responder as próximas questões use as definições a seguir:

Adequação Funcional: É o grau com que o produto provê funções que atendem as necessidades explícitas e implícitas quando usado sobre condições específicas.

Performance: Desempenho relativo à quantidade de recursos usados sobre condições específicas.

Compatibilidade: É o grau com que o produto pode trocar informações com outros produtos e/ou realizar suas funções, enquanto compartilha o mesmo ambiente de hardware e software.

Usabilidade: É o grau com que o produto pode atingir os objetivos específicos de usuários específicos com eficácia, eficiência e satisfação, em um determinado contexto de uso.

Confiabilidade: É o grau com que o produto realiza determinadas funções, sobre condições específicas em um tempo específico.

Segurança: É o grau com que o produto protege dados e informações, de modo que exista graus de acesso a determinadas partes do produto, para determinados tipos e níveis de autorização.

Manutenibilidade: É o grau de eficácia e eficiência com que o produto pode ser modificado pelos seus respectivos mantenedores.

Portabilidade: É o grau com que o produto pode ser transferido de um ambiente de uso para o outro.

Eficácia em uso : Acurácia e completude com que os usuários conseguem atingir seus objetivos.

Eficiência em uso: Recursos gastos em relação a acurácia e completude com que os usuários atingem seus objetivos.

Satisfação do cliente: É o grau com que as necessidades do usuário são satisfeitas pelo uso do produto em um contexto específico.

Inexistência de Risco: É o grau com que o produto mitiga os riscos econômicos, os riscos a vida humana, a saúde, e ao ambiente.

Completude: grau o qual um produto ou sistema pode ser utilizado com eficácia, eficiência, liberdade de riscos, satisfação em todos os contextos de utilização especificados.

1) No momento de escolher um determinado CMS para uma aplicação você leva em conta o Licenciamento?*

Sim

Não

2) Você prefere um CMS software Livre ou software proprietário para desenvolver suas aplicações?*

Software Livre

Software Proprietário

2.1) Justifique sua resposta:

3) No que diz respeito à características funcionais do CMS, quais delas você julga como importantes ou indispensáveis em um CMS? *

Escolha quantas forem necessárias e utilize o campo "Outros" para indicar o que achar importante que esteja faltando na lista (no caso de mais de uma opção separe-as com virgula).

Facilidade de interação do CMS com várias ferramentas. (ex: outras linguagens, bancos de dados, frameworks de programação, etc)

Capacidade de armazenar dados em bancos de dados relacionais.

Boa capacidade para realizar pesquisas ou buscas de conteúdos no próprio site.

Permitir upload de uma grande quantidade de tipos de arquivos Exemplos: (Word, Excel, RTF, PDF, etc).

Permitir download de uma grande quantidade de tipos de arquivos Exemplos: (Word, Excel, RTF, PDF, etc).

Suporte para conteúdos multimídia (ex: vídeos, músicas, etc).

Habilidade para controlar e gerenciar múltiplas versões do mesmo conteúdo.

Geração automática de Interface de Usuário.

Permitir gerenciamento de usuários.

Permitir gerenciamento de arquivos.

- Capacidade de executar Backup dos conteúdos do site.
- Suporte a Fóruns de discussão.
- Suporte a comentários.
- Outro:

4) A quantidade de templates ou plugins disponíveis para um determinado CMS é um fator importante para escolha desse CMS? *

- Sim
- Não

5) No que diz respeito a Usabilidade, quais características você julga como importantes ou indispensáveis na hora de escolher um CMS? *

Escolha quantas características opções forem necessárias e, caso existam características não presentes na lista, inclua-as no campo "outro" separadas por vírgula.

- Interatividade alta.
- Facilidade em criar novos conteúdos.
- Fácil utilização
- Disponibilidade de documentação de fácil uso e entendimento
- Outro:

6) No que diz respeito a Portabilidade, quais características você julga como importante ou indispensáveis na hora de se escolher um CMS para sua aplicação? *

Escolha quantas características opções forem necessárias e, caso existam características não presentes na lista, inclua-as no campo "outro" separadas por vírgula.

- Permitir o gerenciamento de conteúdo em qualquer navegador(Até os de celular)
- Ser multiplataforma (Funcionar independente de sistema operacional) .
- Ser fácil de instalar.
- Outro:

7) No que diz respeito a Manutenibilidade, quais características você julga como importante ou indispensáveis na hora de se escolher um CMS para sua aplicação? *

Escolha quantas características opções forem necessárias e, caso existam características não presentes na lista, inclua-as no campo "outro" separadas por vírgula.

- Fácil configuração para interagir com outros sites.

- Modularidade.
- Boa Taxonomia
- Facilidade em Manutenção.
- Modificabilidade
- Liberdade para alteração de um template específico.
- Reusabilidade
- Outro:

8) No que diz respeito a Segurança, quais características você julga como importante ou indispensáveis na hora de se escolher um CMS para sua aplicação? *

Escolha quantas características opções forem necessárias e, caso existam características não presentes na lista, inclua-as no campo "outro" separadas por vírgula.

- Permitir a restrição de acesso de a um determinado usuário a funcionalidades, de acordo com o seu perfil de acesso.
- Proteção contra SQL injection
- Mecanismo anti phishing
- Mecanismo anti plágio.
- Outro:

9) No que diz respeito à Performance (desempenho) quais características no seu entender que podem prejudicar o desempenho do site na hora de se escolher um CMS para sua aplicação? *

Escolha quantas características opções forem necessárias e, caso existam características não presentes na lista, inclua-as no campo "outro" separadas por vírgula.

- Tempo de carregamento da página.
- Total de Requisições feitas pela página ao se disparar uma determinada ação (funcionalidade, tarefa, atividade, etc).
- Tamanho de Página em KB.
- Outro:

10) A popularidade influencia você a escolher um determinado CMS ?

*

- Sim
- Não

10.1) Justifique sua resposta. *

11) A arquitetura de um CMS é um fator preponderante para a sua escolha ? * Sim

Não

11.1) Justifique a sua resposta *

12) O tamanho da comunidade de pessoas que usam o CMS é um fator preponderante para a escolha de um determinado CMS? *

Sim

Não

12.1) Justifique sua resposta

13) Qua(l)is das características abaixo você julga também importante ter em um CMS, além das já perguntadas? *

Para responder essa questão corretamente veja as definições no topo da página

Compatibilidade

Confiabilidade

Eficácia em uso

Satisfação do Cliente

Inexistência de Riscos

Flexibilidade

Completude

Eficiência em uso

Nenhuma dessas, pois depende da aplicação

14) Além de todas as características apresentadas você gostaria de acrescentar outra que fosse importante para um CMS?

Justifique sua resposta.

****Pergunta Obrigatória***

C.2 Questionário 2 - Validação do Método Proposto

Validação de método para a seleção de sistemas CMS. Questionário desenvolvido por Thiago S. Honorato, estudante de graduação da Engenharia de Software da Universidade de Brasília (UnB) para a finalidade de validar um método proposto para o auxílio da

seleção de sistemas CMS. O questionário se baseia num conjunto de métricas levantadas da ISO 25000, mapeadas de acordo com as características de CMS da literatura. Características validadas por meio de um questionário aplicado anteriormente em comunidades de Sistemas CMS no facebook. O questionário faz parte de um trabalho de conclusão de curso submetido pela Universidade de Brasília.

Para responder o questionário considere as escalas na descrição de cada pergunta e resposta de acordo com as afirmativas.

Grato a Compreensão.

***Obrigatório**

1) No que diz respeito a Adequação Funcional complete: a completude funcional de um determinado CMS é ———. *

Considere como completude funcional o conjunto de funcionalidades necessárias para satisfazer os objetivos da sua aplicação. Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

2) No que diz respeito a Usabilidade complete: as mensagens exibidas pelo CMS são ———. *

Considere como mensagem: mensagens de ajuda, textos informativos e demais textos presentes na página que influenciam tomadas de decisão. Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

3) No que diz respeito a Usabilidade complete :É ——— a documentação disponível auxiliar no aprendizado do CMS. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

4) No que diz respeito a Usabilidade complete: o layout é ——— em um determinado CMS. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

5) No que diz respeito a Segurança complete: o controle de acesso é ——— em um determinado CMS. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante,

4 - *Extremamente Importante*

1 2 3 4

6) No que diz respeito a Segurança, complete: evitar que dados sejam corrompidos em um site feito com CMS é _____ *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

7) No que diz respeito a Portabilidade complete: adaptar o CMS a um determinado ambiente operacional é _____ *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

8) No que diz respeito a Manutenibilidade complete: modificar um CMS para resolver um determinado problema é _____ *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

9) No que diz respeito a Manutenibilidade complete. É _____ o CMS possuir módulos reusáveis. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

10) No que diz respeito a Manutenibilidade complete: o relacionamento entre os componentes é _____ para o funcionamento do módulo. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

11) No que diz respeito a Manutenibilidade complete : testes de sistema são _____ para um CMS. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

12) No que diz respeito a Performance complete : a quantidade de ta-

refas que pode ser realizada por período de tempo é ————. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

13) No que diz respeito a Performance complete: a utilização de memória e outros recursos de hardware é ——— em um CMS. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

14) No que diz respeito a confiabilidade complete: a capacidade de falhas serem identificadas e corrigidas em um projeto com uso de CMS é ————. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

15) No que diz respeito a Confiabilidade complete: um site que utiliza CMS prover disponibilidade 24/7 é ————. *

Considere 24/7 como 24 horas os 7 dias da semana .Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

16) No que diz respeito a Confiabilidade complete: ——— o CMS ser tolerante a falhas. *

Considere como tolerância a falhas a capacidade do CMS de operar conforme o esperado, porém estando sujeito a falhas de hardware/e ou software. Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

17) No que diz respeito a Confiabilidade complete: é ——— o CMS se recuperar após uma falha. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

18) No que diz respeito a Efetividade complete: é ——— um usuário realizar suas atividades e conseguir completa-las de forma correta. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante,

4 - *Extremamente Importante*

1 2 3 4

19) No que diz respeito a Cobertura do Contexto em uso completo: a flexibilidade em uso de determinado CMS é _____. *

Considere que: 1 - Sem Importância, 2 - Pouco Importante, 3 - Muito Importante, 4 - Extremamente Importante

1 2 3 4

APÊNDICE D – GQM - Métricas Definidas

OBS: Os valores descritos para a interpretação das métricas são sugestões. Para uma análise mais precisa de valores é necessária uma investigação mais profunda, seja na literatura, ou por meio de experimentação. A [ISO/IEC-25023 \(2011\)](#) não cita esses valores na descrição de suas métricas.

D.1 Métricas - Adequação Funcional

Tabela 27 – M1 - Cobertura Funcional - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	01
ID	AF-CF-M001
Nome	Cobertura Funcional
Característica	Adequação Funcional
SubCaracterística	Completeness Funcional
Descrição (Questão)	Qual percentual (%) de adequação funcional possui o software?
Função de Medição (fórmula)	A = Numero de perguntas que obtiverem resposta sim. B = Total de perguntas do questionário. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Aplicar questionário. 2. Calcular o valor A. 3. Aplicar a fórmula.
Interpretação	Quanto maior melhor. $0 < x \leq 29$ é pessimo $30 < x \leq 49$ é regular $50 < x \leq 70$ é bom $71 \leq x \leq 100$ é excelente

D.2 Métricas - Usabilidade

Tabela 28 – M2 - Percentual de mensagens claras - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	02
ID	UB-OP-M001
Nome	Percentual de mensagens claras

Característica	Usabilidade
SubCaracterística	Operacionalidade
Descrição (Questão)	Qual percentual (%) de mensagens exibidas pelo sistema que podem ser facilmente entendidas?
Função de Medição (fórmula)	A = Soma de todas as notas das mensagens. B = Número total de mensagens avaliadas. $X = (A/B) * 100\%$.
Método (Forma de coleta)	1. Identificar um conjunto de mensagens no sistema. 2. Elaborar um questionário para validar semanticamente as mensagens. 3. Calcular nota de todas as mensagens válidas. 4. Somar todas as notas de todas as mensagens. 5. Calcular o percentual.
Interpretação	Quanto maior melhor. $0 < x \leq 29$ é péssimo $30 < x \leq 49$ é regular $50 < x \leq 70$ é bom $71 \leq x \leq 100$ é excelente

Tabela 29 – M3 - Documentação Plena - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	03
ID	UB-AP-M001
Nome	Documentação plena
Característica	Usabilidade
SubCaracterística	Aprendizado
Descrição (Questão)	Qual a proporção de funcionalidades descritas na documentação que podem ajudar ou facilitar o uso do software?
Função de Medição (fórmula)	A = Número de funcionalidades descritas na documentação. B = Número total de funcionalidades implementadas. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Logar com o perfil de Admin (Possui todas as funcionalidades). 2. Anotar e contar todas as funcionalidades observadas. 3. Verificar na documentação se as funcionalidades que são descritas. 4. Contar a quantidade de funcionalidades descritas. 5. Fazer calculo da proporção utilizando a fórmula.
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

Tabela 30 – M4 - Percentual de elementos personalizáveis - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	04
ID	US-EST-M001
Nome	Percentual de elementos personalizáveis
Característica	Usabilidade
SubCaracterística	Estética
Descrição (Questão)	Qual percentual (%) de elementos de interface que podem ser personalizados?
Função de Medição (fórmula)	A = Número de elementos de interface que podem ser personalizados. B = número total de elementos de interface. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Escolher uma tela do sistema. 2. Contar o número de elementos da tela escolhida. 3. Contar o número de elementos personalizáveis. 4. Aplicar fórmula.
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

D.3 Métricas - Segurança

Tabela 31 – M5 - Controle de acesso - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	10
ID	SE-CO-M001
Nome	Controle de acesso
Característica	Segurança
SubCaracterística	Confidencialidade
Descrição (Questão)	O quão controlável é o acesso ao sistema?
Função de Medição (fórmula)	A = Número de operações/atividades exclusivas do perfil Autor que foi acessada pelo perfil Assinante. B = Número de tentativas de acesso a operação/atividade exclusiva realizadas. $X = A/B$

Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Criar na aplicação dois usuários distintos. Um usuário comum e um usuário administrador. 2. Dar permissões exclusivas para o usuário administrador. 3. Logar na aplicação como usuário comum e tentar um acesso que só o usuário administrador consiga realizar. 4. Observar se o sistema permite ou não acesso.
Interpretação	<p>Quanto mais próximo de 0 melhor.</p> <p>$X > 2$ é péssimo</p> <p>$1 < X \leq 2$ é regular</p> <p>$0 < X \leq 1$ é bom</p> <p>$X = 0$ é excelente</p>

Tabela 32 – M6 - Prevenção a dados corrompidos - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	11
ID	SEG-INT-M001
Nome	Prevenção a dados corrompidos.
Característica	Segurança
SubCaracterística	Integridade
Descrição (Questão)	Dados corrompidos podem ser evitados?
Função de Medição (fórmula)	<p>A = Número de casos de dados corrompidos que ocorreram ao se executar uma determinada tarefa, ou conjunto de tarefas.</p> <p>B = Número de casos de dados corrompidos previstos.</p> <p>$X = (A/B)$</p>
Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Escolher um conjunto de tarefas em que podem haver dados corrompidos. 2. Contar o número de situações em que podem acontecer dados corrompidos ao se executar as tarefas propostas. 3. Executar as tarefas propostas. 4. Contar o número de situações em que de fato ocorreram dados corrompidos.
Interpretação	<p>Quanto menor melhor.</p> <p>$X > 3$ é péssimo</p> <p>$2 < X < 3$ é regular</p> <p>$1 \leq X \leq 2$ é bom</p> <p>$X = 0$ é excelente.</p>

D.4 Métricas - Portabilidade

Tabela 33 – M7 - Mudança de configuração - Adaptado de ([ISO/IEC-9126-2, 2001](#))

Questão	05
ID	POR-AD-M001

Nome	Mudança de configuração
Característica	Portabilidade
SubCaracterística	Adaptabilidade
Descrição (Questão)	O usuário ou mantenedor pode adaptar o software ao ambiente?
Função de Medição (fórmula)	T = Soma do tempo de operação gasto pelo usuário para completar a adaptação do software ao ambiente do usuário, quando ele tentar instalá-lo ou modificar sua configuração. $X = T$.
Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Selecionar 4 plugins e 1 tema a serem instalados na ferramenta. 2. Realizar três medições do processo de customização da ferramenta com os plugins e temas. 3. Calcular média aritmética das três medições realizadas.
Interpretação	<p>Quanto mais próximo de 0 melhor (mm:ss).</p> <p>$X > 15:00$ é péssimo</p> <p>$10:00 < X \leq 15:00$ é regular</p> <p>$05:00 < X \leq 10:00$ é bom</p> <p>$X \leq 05:00$ é excelente</p>

D.5 Métricas - Manutenibilidade

Tabela 34 – M8 - Complexidade de modificação - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	08
ID	MAN-MOF-M001
Nome	Complexidade de modificação
Característica	Manutenibilidade
SubCaracterística	Modificabilidade
Descrição (Questão)	O mantenedor pode modificar o software para resolver um problema?
Função de Medição (fórmula)	A = Tempo gasto para modificar. B = Número de <i>issues</i> feitas. $X = A/B$
Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Escolher um período de tempo específico (hora, mês, dia, semana). 2. Identificar a quantidade de mudanças feitas na aplicação nesse período de tempo escolhido. (Observar o repositório) 3. Levantar a quantidade de mudanças identificadas no período de tempo proposto.
Interpretação	<p>Quanto menor melhor.</p> <p>$0 < x \leq 3$ é excelente</p> <p>$3 < x \leq 5$ é bom</p> <p>$5 < x \leq 10$ é ruim</p> <p>>10 é péssimo</p>

Tabela 35 – M9 - Condensabilidade - Adaptado de ISO/IEC-25023 (2011)

Questão	06
ID	MAN-MODU-M001
Nome	Condensabilidade
Característica	Manutenibilidade
SubCaracterística	Modularidade
Descrição (Questão)	Qual é a relação entre os componentes presentes em um determinado modulo de um software?
Função de Medição (fórmula)	A = Número de componentes que quando sofrem mudanças não afetam outros componentes. B = Número total de componentes do módulo em questão. $X = A/B$.
Método (Forma de coleta)	1. Identificar um modulo no sistema. 2. Identificar o total de componentes. 3. Identificar quais componentes quando alterados impactam outros componentes. 4. Calcular o resultado.
Interpretação	Quanto mais próximo de 1 melhor. $0 < x \leq 0,29$ é péssimo $0,30 < x \leq 0,59$ é regular $0,6 < x \leq 0,89$ é bom $0,9 \leq x = 1$ é excelente

Tabela 36 – M10 - Percentual de módulos reusáveis - Adaptado de ISO/IEC-25023 (2011)

Questão	07
ID	MAN- REU-M001
Nome	Percentual de módulos reusáveis
Característica	Manutenibilidade
SubCaracterística	Reusabilidade
Descrição (Questão)	Qual percentual (%) de módulos do sistema que podem ser reutilizados?
Função de Medição (fórmula)	A = Número de módulos do sistema que podem ser reutilizados. B = número total de modulos do sistema. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Identificar o número de módulos do sistema. 2. Contar o número de módulos do sistema que podem ser reutilizados. 3. Aplicar fórmula.

Interpretação	<p>Quanto maior melhor.</p> <p>$0 < X \leq 25\%$ é péssimo</p> <p>$25\% < X \leq 50\%$ é regular</p> <p>$50\% < X \leq 75\%$ é bom</p> <p>$75\% < X \leq 100\%$ é excelente.</p>
----------------------	--

Tabela 37 – M11 - Completude de testes de sistema - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	09
ID	MAN-TEST-M001
Nome	Completude de testes de sistema
Característica	Manutenibilidade
SubCaracterística	Testabilidade
Descrição (Questão)	Qual o percentual (%) de testes de sistema que podem ser implementados ?
Função de Medição (fórmula)	<p>A = Número de testes projetados na especificação do sistema.</p> <p>B = Número de testes realizados.</p> <p>$X = (A/B) * 100 \%$</p>
Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Contar o número de testes projetados na especificação do sistema. 2. Contar o número de testes que foram realizados. 3. Aplicar fórmula.
Interpretação	<p>Quanto maior melhor.</p> <p>$0 < X \leq 25\%$ é péssimo</p> <p>$25\% < X \leq 50\%$ é regular</p> <p>$50\% < X \leq 75\%$ é bom</p> <p>$75\% < X \leq 100\%$ é excelente.</p>

D.6 Métricas - Eficiência de Desempenho

Tabela 38 – M12 - Vazão - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	12
ID	PER-TEM-M001
Nome	Vazão
Característica	Eficiência de Desempenho
SubCaracterística	Comportamento do Tempo
Descrição (Questão)	Quantas tarefas podem ser realizadas por período de tempo?
Função de Medição (fórmula)	<p>A = Número de tarefas concluídas.</p> <p>T = Período de tempo observado (minutos) $X = (A/T) * 100 \%$</p>

Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Definir um período de tempo X. 2. Definir um número de tarefas a serem executadas. 3. Observar quantas tarefas são realizadas dentro do período de minutos definidos . 4. Registrar resultados.
Interpretação	<p>Quanto maior melhor.</p> <p>0 < X <= 25% é péssimo</p> <p>25% < X <= 50% é regular</p> <p>50% < X <=75% é bom</p> <p>75% < X <= 100% é excelente.</p>

Tabela 39 – M13 - Utilização de memória - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	13
ID	PER-UR-M001
Nome	Utilização de memória
Característica	Eficiência do Tempo
SubCaracterística	Utilização de Recursos
Descrição (Questão)	Qual a utilização de memória (KB) do sistema ao se realizar uma determinada tarefa?
Função de Medição (fórmula)	<p>A = Quantidade de Memória utilizada ao se realizar uma tarefa.</p> <p>B = Quantidade de memória disponível.</p> <p>$X = (A/B) * 100 \%$</p>
Método (Forma de coleta)	<ol style="list-style-type: none"> 1. Definir uma tarefa X. 2. Verificar a quantidade de memória disponível. 3. Analisar a quantidade de memória gasta ao se realizar a tarefa X .
Interpretação	<p>Quanto maior melhor.</p> <p>0 < X <= 25% é péssimo</p> <p>25% < X <= 50% é regular</p> <p>50% < X <=75% é bom</p> <p>75% < X <= 100% é excelente.</p>

D.7 Métricas - Confiabilidade

Tabela 40 – M14 - Remoção de Falhas - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	14
ID	CON-MAT-M001
Nome	Remoção de Falhas
Característica	Confiabilidade
SubCaracterística	Maturidade

Descrição (Questão)	Qual percentual (%) de falhas corrigidas no projeto?
Função de Medição (fórmula)	A = Número de falhas corrigidas no projeto. B = Número de falhas detectadas. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Contar o número de falhas realizadas durante a execução do projeto de software. 2. Contar o número de falhas encontradas que foram efetivamente corrigidas. 3. Calcular percentual (%)
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

Tabela 41 – M15 - Tempo de Serviço - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	14
ID	CON-DIS-M001
Nome	Tempo de Serviço
Característica	Confiabilidade
SubCaracterística	Disponibilidade
Descrição (Questão)	Qual percentual (%) de disponibilidade do software?
Função de Medição (fórmula)	A = Tempo de disponibilidade efetivamente prestado. B = Tempo de disponibilidade esperado. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Medir o tempo de disponibilidade do software, considerando quedas e eventuais períodos de indisponibilidade. 2. Coletar tempo de disponibilidade real. 3. Aplicar fórmula.
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

Tabela 42 – M16 - Percentual de módulos redundantes instalados - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	14
ID	CON-TOL-M001

Nome	Percentual de módulos redundantes instalados
Característica	Confiabilidade
SubCaracterística	Tolerância a falhas
Descrição (Questão)	Qual percentual (%) de módulos redundantes identificados?
Função de Medição (fórmula)	A = Número de módulos redundantes encontrados. B = Número de módulos existentes no sistema. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Contar o número de módulos do sistema. 2. Contar o número de módulos redundantes.
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

Tabela 43 – M17 - Recuperação após falha - Adaptado de [ISO/IEC-25023 \(2011\)](#)

Questão	14
ID	CON-REC-M001
Nome	Recuperação após falha
Característica	Confiabilidade
SubCaracterística	Capacidade de recuperação
Descrição (Questão)	O software consegue se recuperar após uma falha acontecer?
Função de Medição (fórmula)	T = Tempo de recuperação do software após uma falha ocorrer. $X = T$
Método (Forma de coleta)	1. Forçar uma determinada falha no software em questão. 2. Observar quanto tempo demora para o software se recuperar. 3. Computar o tempo.
Interpretação	Quanto menor melhor. $T > 7 \text{ min}$ é péssimo $5 \text{ min} < X \leq 7 \text{ min}$ é regular $3 \text{ min} < X \leq 5 \text{ min}$ é bom $0 \text{ min} < T \leq 3 \text{ min}$ é excelente.

D.8 Métricas - Efetividade

Tabela 44 – M18 - Percentual de tarefas completas - Adaptado de [ISO/IEC-25022 \(2012\)](#)

Questão	16
ID	EFI-EFI-M001
Nome	Percentual de tarefas completas
Característica	Efetividade
SubCaracterística	Efetividade
Descrição (Questão)	Qual percentual (%) de tarefas são completadas de forma correta?
Função de Medição (fórmula)	A = Número de tarefas completadas de forma correta. B = Número de tarefas tentadas. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Escolher um número de tarefas para serem executadas. 2. Executar as tarefas. 3. Contar o número de tarefas que foram completadas. 4. Aplicar fórmula.
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

D.9 Métricas - Cobertura de Contexto

Tabela 45 – M19 - Características flexíveis - Adaptado de [ISO/IEC-25022 \(2012\)](#)

Questão	15
ID	COR-FLEX-M001
Nome	Características flexíveis
Característica	Cobertura de Contexto
SubCaracterística	Flexibilidade
Descrição (Questão)	Qual percentual (%) de características de design flexível possui o produto em questão?
Função de Medição (fórmula)	A = Número de características de design que podem ser flexíveis. B = Número total de características de design. $X = (A/B) * 100 \%$
Método (Forma de coleta)	1. Encontrar na documentação o total de características de design . 2. Identificar quais características de design podem ser flexíveis. 3. Aplicar fórmula.
Interpretação	Quanto maior melhor. $0 < X \leq 25\%$ é péssimo $25\% < X \leq 50\%$ é regular $50\% < X \leq 75\%$ é bom $75\% < X \leq 100\%$ é excelente.

APÊNDICE E – Suporte para as métricas definidas

E.1 Completude Funcional

O questionário definido a seguir foi baseado no questionário aplicado no estudo de [Nath e Arora \(2010\)](#) com uma mescla de características sugeridas pelos especialistas na seção 6.2.2.

Tabela 46 – Questionário checklist para a execução da métrica de Completude Funcional.

#	Característica	Sim	Não
1	O CMS gera interface de usuário automaticamente?		
2	O Usuário precisa de login para ter acesso as funcionalidades do CMS?		
3	O CMS fornece opção de recuperar senha? Caso a senha seja esquecida?		
4	O CMS notifica o usuário com um e-mail ao criar uma conta?		
5	O CMS realiza backup?		
6	O CMS gerencia o conteúdo?		
7	O CMS oferece suporte a conteúdos multimídia (Vídeos, músicas e imagens) ?		
8	O CMS permite troca de temas?		
9	O CMS é multiplataforma?		
10	O CMS permite suporte a vários bancos de dados?		
11	O CMS faz geração automática código?		
12	O CMS permite gerenciar múltiplos usuários?		
13	O CMS permite realizar upload de arquivos?		
14	CMS permite realizar download de arquivos?		
15	O CMS permite gerenciar arquivos?		
16	O CMS oferece mecanismo anti plágio?		
17	O CMS oferece suporte a comentários?		
18	O CMS oferece suporte para a criação de fóruns?		
19	O CMS oferece mecanismos de busca?		
20	O CMS oferece para o usuário uma localização no site?		
21	O CMS oferece suporte para e-commerce?		

22	O CMS pode ser estendido?		
-----------	---------------------------	--	--

E.2 Operacionalidade

Para a métrica de operacionalidade foi feito um questionário simples baseado em três princípios de usabilidade de Nielsen (Estética e Design, Consistência e Padrões e Visibilidade do status do sistema) como aplicado na pesquisa de (NIELSEN; LANDAUER, 1993 apud ALVES; PIRES, 2002).

O questionário é apresentado a seguir acompanhado com o critério de atribuição de nota a cada pergunta do questionário.

Questionário

1. Corretude gramatical da mensagem (Pontos, vírgulas, acentos, palavras erradas). Nota: Vai de 1 a 5.

Tabela 47 – Parâmetros de corretude gramatical

	0 Erros	1 Erro	2 Erros	3 Erros	4 Erros ou mais
Nota	5	4	3	2	1

2. Contraste de cores entre cor do texto e cor de fundo. Nota: Vai de 1 a 5.

Uma sugestão de software para ser usado nesta etapa é o *Colour Contrast Analyzer (CCA)*¹. O CCA faz análise de contraste de cores de elementos em uma determinada tela. O CCA compara o contraste de cores entre dois pixels da tela, e gera o resultado da comparação, informando quais critérios foram satisfatórios e quais falharam. Nesse sentido, a nota para essa pergunta é calculada da seguinte forma:

$$\text{Nota} = [(\text{Total de critérios satisfatórios}) / (\text{Total de critérios})] * 5$$

Obs: Total de critérios na ferramenta CCA é sempre igual a 4, logo:

$$\text{Nota} = [(\text{Total de critérios satisfatórios}) / 4] * 5$$

Obs2: - Se a nota for um número decimal de até 0,5, ele será arredondado para baixo. Ex: 4,5 = 4.

- Se a nota for um número decimal maior do que 0,5, ele será arredondado para cima. Ex: 4,6 = 5.

- Se a nota for igual a 0 (Total de critérios satisfatórios = 0), o número será arredondado para 1, pois essa é a menor nota possível.

¹Disponível em: "<http://www.paciellogroup.com/resources/contrastanalyser/>"

3. Textos curtos. Nota: Vai de 1 a 5.

Obs: X = Número de palavras.

Tabela 48 – Textos curtos

	$X \leq 3$	$3 < X \leq 4$	$4 < X \leq 5$	$5 < X \leq 6$	$X \geq 7$
Nota	5	4	3	2	1

4. Terminologia usada em relação a tarefa sendo realizada. Nota: É 1 ou 5.

Tabela 49 – Terminologia

	Bom	Ruim
Nota	5	1

Nota final de cada mensagem:

A nota final de cada mensagem é calculada pela seguinte fórmula:

$$\text{Nota final mensagem} = (\text{Soma das notas das questões})/20$$

Nota final da métrica:

A nota final da métrica é calculada pela seguinte fórmula:

$$\text{Nota final} = [(\text{Soma total de todas as notas das mensagens}) / (\text{Numero total de mensagens avaliadas})] * 100$$

APÊNDICE F – Cargas fatoriais

A Tabela 50 mostra as cargas fatoriais divididas entre os seis fatores principais. Os valores em negrito representam as cargas fatoriais mais relevantes para as questões. A partir disso, observou-se que:

- Para o Fator 1 os conceitos participantes são: Estética, Modificabilidade, Modularidade, Testabilidade, Comportamento do Tempo, Utilização de Recursos, Maturidade, Recuperabilidade e Efetividade.
- Para o Fator 2 os conceitos participantes são: Confidencialidade, Disponibilidade e Tolerância a Falhas.
- Para o Fator 3 os conceitos participantes são: Operacionalidade, Reusabilidade.
- Para o Fator 4 os conceitos participantes são: Completude Funcional, Adaptabilidade, Flexibilidade.
- O Fator 5 tem somente um único conceito participante que é o Aprendizado.
- O Fator 6 tem somente um único conceito participante que é a Integridade.

Tabela 50 – Cargas fatoriais para seis fatores

Matriz de Componentes - seis fatores						
Questão - Conceito	Componentes					
	1	2	3	4	5	6
Q1 - Completude Funcional	0,237	-0,505	0,292	0,320	0,019	-0,116
Q2 - Operacionalidade	0,340	-0,290	0,468	0,339	-0,113	-0,166
Q3 - Aprendizado	0,265	0,203	-0,124	0,290	0,636	0,144
Q4 - Estética	0,662	-0,125	-0,329	-0,156	-0,004	-0,050
Q5 - Confidencialidade	0,075	0,644	0,122	0,343	0,141	-0,411
Q6 - Integridade	0,090	0,421	0,407	0,123	0,058	0,618
Q7- Adaptabilidade	0,353	-0,100	-0,117	0,392	-0,475	0,249
Q8 - Modificabilidade	0,389	0,228	0,154	-0,157	-0,571	0,135
Q9 - Reusabilidade	0,423	0,115	0,695	-0,216	0,043	-0,134
Q10 - Modularidade	0,539	0,114	0,271	-0,270	0,136	-0,049
Q11 - Testabilidade	0,640	-0,114	-0,245	-0,225	0,072	-0,158
Q12 - Comportamento do Tempo	0,652	-0,153	-0,294	-0,020	-0,075	-0,045
Q13 - Utilização de Recursos	0,548	-0,079	-0,084	-0,420	0,158	0,245

Q14 - Maturidade	0,476	0,047	0,203	-0,251	0,349	-0,044
Q15 - Disponibilidade	0,279	0,678	0,018	0,049	-0,236	-0,122
Q16 - Tolerância a Falhas	0,377	0,529	-0,317	0,186	-0,083	-0,289
Q17 - Recuperabilidade	0,702	0,016	-0,084	0,138	-0,189	0,177
Q18 - Efetividade	0,410	-0,545	0,121	0,299	0,035	-0,173
Q19 - Flexibilidade	0,355	0,041	-0,171	0,454	0,321	0,291

Tabela 51 – Cargas fatoriais para cinco fatores

Questão - Conceito	Matriz componentes - cinco fatores				
	Componentes				
	1	2	3	4	5
Q1 - Completude Funcional	0,237	-0,505	0,292	0,320	0,019
Q2 - Operacionalidade	0,340	-0,290	0,468	0,339	-0,113
Q3 - Aprendizado	,265	0,203	-0,124	0,290	0,636
Q4 - Estética	0,662	-0,125	-0,329	-0,156	-0,004
Q5 - Confidencialidade	0,075	0,644	0,122	0,343	0,141
Q6 - Integridade	0,090	0,421	0,407	0,123	0,058
Q7- Adaptabilidade	0,353	-0,100	-0,117	0,392	-0,475
Q8 - Modificabilidade	0,389	0,228	0,154	-0,157	-0,571
Q9 - Reusabilidade	0,423	0,115	0,695	-0,216	0,043
Q10 - Modularidade	0,539	0,114	0,271	-0,270	0,136
Q11 - Testabilidade	0,640	-0,114	-0,245	-0,225	0,072
Q12 - Comportamento do Tempo	0,652	-0,153	-0,294	-0,020	-0,075
Q13 - Utilização de Recursos	0,548	-0,079	-0,084	-0,420	0,158
Q14 - Maturidade	0,476	0,047	0,203	-0,251	0,349
Q15 - Disponibilidade	0,279	0,678	0,018	0,049	-0,236
Q16 - Tolerância a Falhas	0,377	0,529	-0,317	0,186	-0,083
Q17 - Recuperabilidade	0,702	0,016	-0,084	0,138	-0,189
Q18 - Efetividade	0,410	-0,545	0,121	0,299	0,035
Q19 - Flexibilidade	0,355	0,041	-0,171	0,454	0,321

De forma semelhante a Tabela 50 a Tabela 51 não evidencia um agrupamento de conceitos que esteja de acordo com a literatura.

Tabela 52 – Cargas fatoriais para quatro fatores

Questão - Conceito	Matriz de componentes quatro - Fatores			
	Componentes			
	1	2	3	4

Q1 - Completude Funcional	0,237	-0,505	0,292	0,320
Q2 - Operacionalidade	0,340	-0,290	0,468	0,339
Q3 - Aprendizado	0,265	0,203	-0,124	0,290
Q4 - Estética	0,662	-0,125	-0,329	-0,156
Q5 - Confidencialidade	0,075	0,644	0,122	0,343
Q6 - Integridade	0,090	0,421	0,407	0,123
Q7- Adaptabilidade	0,353	-0,100	-0,117	0,392
Q8 - Modificabilidade	0,389	0,228	0,154	-0,157
Q9 - Reusabilidade	0,423	0,115	0,695	-0,216
Q10 - Modularidade	0,539	0,114	0,271	-0,270
Q11 - Testabilidade	0,640	-0,114	-0,245	-0,225
Q12 - Comportamento do Tempo	0,652	-0,153	-0,294	-0,020
Q13 - Utilização de Recursos	0,548	-0,079	-0,084	-0,420
Q14 - Maturidade	0,476	0,047	0,203	-0,251
Q15 - Disponibilidade	0,279	0,678	0,018	0,049
Q16 - Tolerância a Falhas	0,377	0,529	-0,317	0,186
Q17 - Recuperabilidade	0,702	0,016	-0,084	0,138
Q18 - Efetividade	0,410	-0,545	0,121	0,299
Q19 - Flexibilidade	0,355	0,041	-0,171	0,454

A Tabela 52 ainda não apresenta um agrupamento compatível com a literatura. Um fato importante que ocorreu nesta etapa é que a característica de aprendizado não apresenta valores relevantes de carga fatorial para fazer parte de nenhum fator.

Tabela 53 – Cargas fatoriais para três fatores

Matriz de Componentes três fatores			
Questão - Conceito	Componentes		
	1	2	3
Q1 - Completude Funcional	0,237	-0,505	0,292
Q2 - Operacionalidade	0,340	-0,290	0,468
Q3 - Aprendizado	0,265	0,203	-0,124
Q4 - Estética	0,662	-0,125	-0,329
Q5 - Confidencialidade	0,075	0,644	0,122
Q6 - Integridade	,090	0,421	0,407
Q7- Adaptabilidade	0,353	-0,100	-0,117
Q8 - Modificabilidade	0,389	0,228	0,154
Q9 - Reusabilidade	0,423	0,115	0,695
Q10 - Modularidade	0,539	0,114	0,271
Q11 - Testabilidade	0,640	-0,114	-0,245

Q12 - Comportamento do Tempo	0,652	-0,153	-0,294
Q13 - Utilização de Recursos	0,548	-0,079	-0,084
Q14 - Maturidade	0,476	0,047	0,203
Q15 - Disponibilidade	0,279	0,678	0,018
Q16 - Tolerância a Falhas	0,377	0,529	-0,317
Q17 - Recuperabilidade	0,702	0,016	-0,084
Q18 - Efetividade	0,410	-0,545	0,121
Q19 - Flexibilidade	0,355	0,041	-0,171

A Tabela 53 que mostra a divisão de conceitos para três fatores ainda não apresenta uma divisão equivalente com a literatura. Nesta etapa observou-se que os conceitos completude funcional e aprendizado são irrelevantes para estes fatores, pois não possuem carga fatorial $\geq 0,3$.

Tabela 54 – Cargas fatoriais para dois fatores

Matriz de Componentes dois fatores		
Questão - Conceito	Componentes	
	1	2
Q1 - Completude Funcional	0,237	-0,505
Q2 - Operacionalidade	0,340	-0,290
Q3 - Aprendizado	0,265	0,203
Q4 - Estética	0,662	-0,125
Q5 - Confidencialidade	0,075	0,644
Q6 - Integridade	0,090	0,421
Q7- Adaptabilidade	0,353	-0,100
Q8 - Modificabilidade	0,389	0,228
Q9 - Reusabilidade	0,423	0,115
Q10 - Modularidade	0,539	0,114
Q11 - Testabilidade	0,640	-0,114
Q12 - Comportamento do Tempo	0,652	-0,153
Q13 - Utilização de Recursos	0,548	-0,079
Q14 - Maturidade	0,476	0,047
Q15 - Disponibilidade	0,279	0,678
Q16 - Tolerância a Falhas	0,377	0,529
Q17 - Recuperabilidade	0,702	0,016
Q18 - Efetividade	0,410	-0,545
Q19 - Flexibilidade	0,355	0,041

A Tabela 54 representa a distribuição de conceitos em apenas dois fatores. Porém, esta divisão ainda não bate com a literatura, pois se o Fator 1 fosse chamado de qualidade do produto e o Fator 2 de qualidade em uso, por exemplo, seriam constatado conceitos inválidos. Os conceitos inválidos para o Fator 1 seriam a efetividade e a flexibilidade, pois são conceitos de qualidade em uso. Já os conceitos presentes no Fator 2 são todos de qualidade do produto, o que geraria uma divergência quanto as definições da SQuaRE.