



**Universidade de Brasília - UnB  
Faculdade UnB Gama - FGA  
Curso de Engenharia De Software**

**Avaliação de Softwares Legados com Base em  
Métricas: em um Órgão Público Federal**

**Autor: Larissa Rodrigues Gomes  
Orientadora: Elaine Venson**

**Brasília, DF  
2015**



**LARISSA RODRIGUES GOMES**

**Avaliação de Softwares Legados com Base em Métricas:  
em um Órgão Público Federal**

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.

Orientadora: Elaine Venson

**Brasília, DF  
2015**

## CIP – Catalogação Internacional da Publicação\*

Gomes, Larissa Rodrigues.

Avaliação de Softwares Legados com Base em Métricas em um Órgão Público Federal/ Larissa Rodrigues Gomes. Brasília: UnB, 2015. 77 p. : il. ; 29,5 cm.

Monografia (Graduação) – Universidade de Brasília Faculdade do Gama, Brasília, 2015. Orientação: Elaine Venson.

1. Software Legado. 2. Manutenção de Software. 3. Métrica de Software 3 I. Venson, Elaine. II. Título.

CDU Classificação



## **AVALIAÇÃO DE SOFTWARES LEGADOS COM BASE EM MÉTRICAS: EM UM ÓRGÃO PÚBLICO FEDERAL**

**Larissa Rodrigues Gomes**

Monografia submetida como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software da Faculdade UnB Gama - FGA, da Universidade de Brasília, em (data da aprovação 02/12/2015) apresentada e aprovada pela banca examinadora abaixo assinada:

---

**Prof.: Elaine Venson, UnB/ FGA**  
Orientadora

---

**Prof: Ricardo Ajax Dias Kosloski, UnB/ FGA**  
Membro Convidado

---

**Prof: Dr. Luiz Carlos Miyadaira, MC**  
Membro Convidado

Brasília, DF  
2015

"Mesmo que o futuro lhe pareça distante, ele está começando neste exato momento".  
(Mattie J.T. Stepanek)

## **AGRADECIMENTOS**

Agradeço primeiramente a minha mãe, Adriane Rodrigues, e a minha tia, Alessandra Rodrigues, pelo apoio e incentivo em todos esses anos de graduação. E também a toda a minha família por acreditar, incentivar e por ter sido meu porto seguro. Sem vocês essa jornada não seria possível.

Agradeço a todos os professores que nos acompanham durante a graduação e a minha orientadora professora Elaine Venson, que dedicou-se à orientação deste trabalho de maneira notável, incentivando e apoiando.

Agradeço ao Alípio Valadares por, ter me acompanhado durante todas as etapas da minha graduação, por todo o companheirismo e paciência (sobretudo nos dias que antecederam a entrega deste trabalho) e por compartilhar dos meus sonhos.

Quero agradecer também ao Abisaí Nunes e ao curso CIB que me auxiliou na construção deste trabalho proporcionando uma melhor observação e interpretação dos artigos e textos utilizados, além de uma melhor sintetização e exposição das ideias apresentadas.

Muito obrigada!

## RESUMO

A contratação de serviços de Tecnologia da Informação (TI) é uma prática comum em órgãos públicos. A terceirização envolve um elevado consumo de recursos públicos e uma alta complexidade de gestão. No cenário público federal brasileiro, o serviço de manutenção de software é terceirizado. Softwares legados são sistemas já desenvolvidos e que executam tarefas importantes para a organização em que eles estão inseridos. Para que os softwares legados se mantenham alinhados às expectativas da organização é necessário que sejam constantemente modificados, o que acarreta esforço e custo com atividades de manutenção e por isso é necessário que a organização tenha uma visão clara sobre o atual estado de seus sistemas legados. A falta de conhecimento sobre aspectos dos sistemas legados motiva uma análise aprofundada dos sistemas. Dessa forma esse trabalho visa aplicar métricas nos softwares legados a partir da utilização do método GQM (*goal question metric*) para caracterizar três aspectos principais dos softwares legados que são: qualidade interna do código, documentação e registros de manutenção corretiva. Essa caracterização visa identificar as fraquezas e proporcionar dados quantitativos e qualitativos, que possibilitem uma maior visão do gestor de TI sobre seus softwares legados.

**Palavras-chave:** Manutenção de software; Softwares Legados, Métricas de Software.

## ABSTRACT

Hiring of Information Technology (IT) services is a common practice in public agencies. Outsourcing involves a high consumption of public resources and a highly complex management. In the Brazilian federal public arena, the software maintenance service is outsourced. Legacy software systems are already developed and performing important tasks for the organization in which they are inserted. For legacy software remain aligned with the organization's expectations they must be constantly modified, resulting effort and cost of maintenance activities and therefore it is necessary that the organization has a clear view about the current state of their legacy systems. The lack of knowledge about aspects of legacy systems motivates a thorough examination of the systems. Thus this work aims to apply metrics in legacy software from using the GQM method (goal question metric) to characterize three main aspects of legacy software that are internal code quality, documentation and corrective maintenance records. This characterization is to identify weaknesses and provide quantitative and quantitative data that enable greater insight into the IT manager about your legacy software.

**Keywords:** *Software maintenance; Legacy software; Software Metrics.*



## LISTA DE FIGURAS

Figura 1: Seleção metodológica. Fonte autor.....	21
Figura 2: Plano metodológico. Fonte: Autor.....	23
Figura 3: Avaliação de sistemas legados. Fonte:(SOMERVILLE, 2011).....	32
Figura 4: Matriz de decisão. (JACOBSON, 1991; adaptado). ....	38
Figura 5: GQM. Fonte: (BASILI, 1994, adaptado) .....	44
Figura 6: Etapas de execução do estudo de caso. Fonte: Autor.....	53
Figura 7: Fluxo de atividades realizadas. Fonte autor.....	60
Figura 8: Tamanho de sistemas em linhas de código. Fonte: Autor.....	76
Figura 9: Sistemas com gravidade alta. Fonte: Autor.....	83
Figura 10: Relatório de documentações presentes nos sistemas. Fonte: Autor.....	89
Figura 11: Representação dos valores das métricas para cada sistema. Fonte: Autor. .....	92
Figura 12: Porcentagem de sistemas quanto a documentação do modelo de banco de dados. Fonte: Autor.....	94
Figura 13: Percentual de sistemas e completude da documentação de regras de negócio. Fonte: Autor.....	95
Figura 14: Percentual de sistemas com os requisitos funcionais. Fonte: Autor.....	96
Figura 15: Número de manutenções corretivas por sistema. Fonte: Autor. ....	103
Figura 16: Níveis de criticidade dos sistemas: Fonte: Autor.....	105
Figura 17: Representação das classificações das manutenções corretivas. Fonte: Autor.....	108
Figura 18: Percentual de defeitos do sistema REPNBL. Fonte: Autor. ....	109
Figura 19: Percentual de defeitos por tipo do sistema CPROD. Fonte: Autor.....	110
Figura 20: Percentual de defeitos por tipo do sistema SEI. Fonte: Autor. ....	111
Figura 21: Percentual de defeitos por tipo do sistema CADSEI. Fonte: Autor.....	111
Figura 22: Percentual de defeitos por tipo do sistema OTRS. Fonte: Autor.....	112
Figura 23: Percentual de defeitos por tipo do sistema RADCOM. Fonte: Autor.....	113
Figura 24: Divisão de quadrantes para análise. Fonte: Autor.....	118
Figura 25: Representação da documentação e do número de manutenções corretivas. Fonte: Autor. ....	121
Figura 26: Representação dos quadrantes. Fonte: Autor.....	122
Figura 27: Representação da qualidade e os número de manutenção corretiva dos sistemas. Fonte: Autor. ....	124
Figura 28: Divisão do quadrantes. Fonte: Autor.....	125
Figura 29: Representação dos valores de qualidade e documentação dos sistemas. Fonte: Autor.....	128
Figura 30: Registro de manutenção corretiva sistema REPNBL. Fonte: Autor. ....	128
Figura 31: Registro de manutenção corretiva sistema CPROD. Fonte: Autor.....	129
Figura 32: Registro de manutenção corretiva sistema SEI. Fonte: Autor.....	129
Figura 33: Registros de manutenção corretiva sistema CADSEI. Fonte: Autor.....	130
Figura 34: Registro de manutenção corretiva sistema OTRS. Fonte: Autor.....	130
Figura 35: Registro de manutenção corretiva sistema RADCOM. Fonte: Autor.....	131
Figura 36: Matriz de decisão sobre os sistemas. Fonte: (adaptado, JACOBSON; 1991). ....	134

## LISTA DE TABELAS

Tabela 1:Objetivos estratégicos específicos ao PETI. Fonte: (BRASIL, 2015, adaptado).....	55
Tabela 2: Análise SWOT da CGTI do Órgão. Fonte: (BRASIL, 2015, adaptado).....	56
Tabela 3: Questões definidas para atender o objetivo de medição. Fonte autor.....	64
Tabela 4: Questão 1. Avaliação da qualidade. Fonte: (SONAR, 2015).....	65
Tabela 5: Questão 2. Avaliação da qualidade. Fonte: (SONAR, 2015).....	65
Tabela 6: Questão 3. Avaliação da qualidade. Fonte: (SONAR, 2015).....	66
Tabela 7: Questão 4. Avaliação da Qualidade. Fonte: (SONAR, 2015).....	67
Tabela 8: Questão 5. Avaliação da Qualidade. Fonte: (SONAR, 2015).....	68
Tabela 9: Questões definidas para o objetivo 2. Fonte: autor. ....	69
Tabela 10: Questão 1. Avaliação da documentação do sistema. Fonte: (RAMOS, 2004).....	69
Tabela 11: Questões definidas para o objetivo 3. Fonte: autor. ....	71
Tabela 12: Questão 1. Classificação de defeitos. Fonte: Autor.....	71
Tabela 13: Questão 2. Classificação de defeitos. Fonte: Autor.....	71
Tabela 14: Lista de sistemas ativos. Fonte: autor. ....	72
Tabela 15: Primeira métrica, tamanho do sistema em linhas de código. Fonte: Autor. ....	74
Tabela 16: Segunda métrica do primeiro objetivo de medição. Fonte: Autor. ....	76
Tabela 17: Critérios de classificação de duplicação do sistema. Fonte: Autor. ....	78
Tabela 18: Sistemas pertencentes a cada classificação. Fonte: Autor. ....	78
Tabela 19: Terceira questão do objetivo de medição 1. Fonte: Autor. ....	79
Tabela 20: Valores de complexidade dos sistemas. Fonte: (adaptado, Schaidt, Rodrigues; 2015).....	80
Tabela 21: Sistemas pertencentes as classificações de complexidade. Fonte: Autor. ....	80
Tabela 22: Quarta questão do objetivo de medição 1. Fonte: Autor. ....	81
Tabela 23: Quinta questão do objetivo de medição 1. Fonte: Autor.....	84
Tabela 24: Valores de referência para cobertura de testes. Fonte: (Adaptado, Schaidt, Rodrigues; 2015).....	85
Tabela 25: Relação de sistemas com suas classificações. Fonte: Autor. ....	85
Tabela 26: Relação dos sistemas com suas documentações existentes. Fonte: Autor. ....	87
Tabela 27: Nome dos sistemas que possuem a documentação mínima. Fonte: Autor. ....	89
Tabela 28: Relação de sistemas que possuem alguma documentação. Fonte: Autor. ....	90
Tabela 29: Relação de nome de sistemas sem documentação. Fonte: Autor. ....	90
Tabela 30: Primeira questão do objetivo de medição 2. Fonte: Autor. ....	91
Tabela 31: Classificação do sistemas em relação a sua documentação. Fonte: Autor. ....	93
Tabela 32: Classificação dos sistemas quanto aos seus requisitos funcionais. Fonte: Autor.....	94
Tabela 33: Relação de sistemas com a documentação de requisitos funcionais. Fonte: Autor.....	95
Tabela 34: Descrição de aspectos do conteúdo das documentações. Fonte: Autor. ....	97
Tabela 35: Primeira questão do objetivo de medição 3. Fonte: Autor.....	102

Tabela 36: Valor de severidade de acordo com o número de manutenção corretiva. Fonte: Autor.....	104
Tabela 37: Sistemas e s relação de severidades. Fonte: Autor. ....	104
Tabela 38: Segunda questão objetivo de medição 3. Fonte: Autor. ....	106
Tabela 39: Gravidade dos erros e pesos utilizados. Fonte: Autor. ....	114
Tabela 40: Relação de sistemas com RCI, índice de documentação e número de manutenções. Fonte: Autor. ....	116
Tabela 41:Relação entre o número de manutenções corretivas e índice de documentação. Fonte: Autor. ....	119
Tabela 42: Relação dos sistemas pertencentes a cada quadrante. Fonte: Autor. ..	119
Tabela 43: Interpretação dos valores pertencentes a cada quadrante. Fonte: Autor. .....	123
Tabela 44: Relação de sistemas pertencentes a cada quadrante. Fonte: Autor. ....	123
Tabela 45: Interpretação dos valores do quadrantes. Fonte: Autor.....	126
Tabela 46: Sistemas pertencentes a cada quadrante. Fonte: Autor.....	126
Tabela 47: Valor da criticidade dos sistemas. Fonte: Autor. ....	132
Tabela 48: Valores do RCI e criticidade de cada sistema. Fonte: Autor. ....	133
Tabela 49: Interpretação dos valores do quadrantes. Fonte: Autor.....	135
Tabela 50: Decisão a ser tomada para cada sistema. Fonte: Autor.....	136

## LISTA DE QUADROS

Quadro 1: Objetivo de negócio. Fonte: Autor. ....	63
Quadro 2: Objetivo 1: Avaliação da qualidade do código do sistema legado. Fonte autor. ....	64
Quadro 3: Avaliação das documentações. Fonte: Autor. ....	69
Quadro 4: Objetivo 3. Fonte: autor. ....	70

**LISTA DE EQUAÇÕES**

Equação 1: Formula do RCI. Fonte: (Adaptado, Sonar, 2015).....	115
Equação 2: Formula do índice de documentação. Fonte: Autor.....	116

## LISTA DE ABREVIATURAS

<b>APF</b>	Administração Pública Federal
<b>TI</b>	Tecnologia da Informação
<b>GQM</b>	Goal/ Question Metric
<b>PETI</b>	Plano Estratégico de Tecnologia da Informação
<b>PDTI</b>	Plano Diretor de Tecnologia da Informação
<b>CADSEI</b>	Cadastro de Usuários Externos
<b>CEL</b>	Comissão Especial de Licitação
<b>CPROD</b>	Controle de processo e Documentos
<b>SCV</b>	Controle de visitantes
<b>DSCOM</b>	Dados do setor de comunicação
<b>GEO</b>	Geoserve
<b>GPD</b>	Gestão de Processos e documentos
<b>OTRS</b>	Open Source Tiket Request System
<b>RADTEC</b>	Radiofusão técnica
<b>ASPAR</b>	Sistema de acessória parlamenta
<b>SICAF</b>	Sistema de acompanhamento de radiofusão
<b>SAF</b>	Sistema de arrecadação do fundel
<b>SCCA</b>	Sistema de compromisso cerimonial e agenda
<b>SICAP</b>	Sistema de Consulta
<b>SCP</b>	Sistema de consulta de processo
<b>SGD</b>	Sistema de gerenciamento de documentos
<b>SISGESAC</b>	Sistema de governo eletrônico registro e atendimento ao cidadão
<b>SISPA</b>	Sistema de pastas Funcionais, Jurídicas e Técnicas
<b>SIP</b>	Sistema de Permissão
<b>RADCOM</b>	Sistema de radiofusão comunitária
<b>SRH</b>	Sistema de recursos humanos
<b>SRAPD</b>	Sistema de registro e análise de proposições de documentos
<b>SEI</b>	Sistema Eletrônico de informação
<b>RCI</b>	<i>Rules Compliance Metric</i>

## SUMÁRIO

<b>1. INTRODUÇÃO .....</b>	<b>16</b>
1.1. CONTEXTO .....	16
1.2. PROBLEMA .....	17
1.3. OBJETIVO .....	19
1.4. METODOLOGIA .....	19
1.4.1 Estudo de Caso .....	21
1.4.2 Fases do Desenvolvimento.....	23
1.5. ORGANIZAÇÃO DO TRABALHO.....	24
<b>2. MANUTENÇÃO DE SOFTWARES LEGADOS.....</b>	<b>25</b>
2.1. MANUTENÇÃO .....	25
2.2. TIPOS DE MANUTENÇÃO.....	26
2.3. MANUTENIBILIDADE .....	26
2.4. MANUTENÇÃO E CUSTO .....	28
2.5. SOFTWARES LEGADOS.....	30
2.6. ENGENHARIA REVERSA .....	33
2.7. REENGENHARIA .....	35
2.7.1 Processo de Reengenharia .....	36
2.7.2 Reengenharia Aplicada.....	37
<b>3. MEDIÇÃO E MÉTRICAS.....</b>	<b>40</b>
3.1. MEDIÇÃO DE SOFTWARE.....	40
3.2. MÉTRICAS DE SOFTWARE .....	40
3.3. PROCESSO DE MEDIÇÃO DE SOFTWARE.....	42
3.4. GQM ( <i>Goal Question Metric</i> ) .....	43
3.5. VERIFICAÇÃO DE SOFTWARE .....	45
3.6. INSPEÇÃO DE SOFTWARE.....	45
3.7. ANÁLISE ESTÁTICA DE CÓDIGO.....	46
3.8. FERRAMENTAS DE ANÁLISE ESTÁTICA .....	47
3.9. SONARQUEB.....	47
<b>4. MATERIAIS E MÉTODOS.....</b>	<b>50</b>
4.1. PLANO DE COLETA DE MÉTRICAS .....	50
4.2. ESTUDO DE CASO.....	52
4.2.1 Planejamento e Desenho do Estudo de Caso .....	53
4.2.1.1. Caracterização do Objeto de Estudo .....	53
4.2.1.2. Fundamentação do trabalho .....	57
4.2.2 Coleta de dados.....	57
4.2.2.1 Análise Documental .....	58
4.2.2.2 Entrevista Informal .....	58
4.2.2.3 Métricas .....	59
4.2.3 Análise dos dados.....	61
4.2.3.1 Validação dos Dados .....	61
4.2.4 Redação do Estudo .....	62
<b>5 AVALIAÇÃO DOS SISTEMAS LEGADOS .....</b>	<b>63</b>
5.1 APLICAÇÃO DO GQM .....	63
5.1.1 Coleta e Análise dos dados .....	72
5.1.1.1 Objetivo de medição 1: Avaliação da qualidade interna do código do sistema legado.....	73
5.1.1.2 Objetivo de medição 2: Avaliação da documentação disponível .....	86
5.1.1.3 Objetivo de medição 3: análise da quantidade de requisições de manutenção corretiva registradas 101	
<b>6 COMPARAÇÃO E ANÁLISE DOS DADOS.....</b>	<b>114</b>
6.1 COMPARAÇÃO ENTRE OS DADOS .....	114
6.1.1 Manutenção corretiva e documentação .....	118
6.1.2 Manutenção corretiva e qualidade .....	121
6.1.3 Qualidade e Documentação .....	125
6.2 REGISTROS DE MANUTENÇÃO CORRETIVA .....	128
6.3 MATRIZ DE DECISÃO PARA SOFTWARES LEGADOS.....	131
<b>7 CONCLUSÕES E TRABALHOS FUTUROS.....</b>	<b>137</b>
<b>Referências Bibliográficas .....</b>	<b>141</b>

## 1. INTRODUÇÃO

Este primeiro capítulo aborda o contexto em que este trabalho está inserido, o problema abordado, sua justificativa, objetivos propostos, resultados esperados e os capítulos referentes ao desenvolvimento do mesmo.

### 1.1. CONTEXTO

Na Administração Pública Federal (APF), a manutenção de softwares é realizada por meio da terceirização utilizando como meio a atividade de contratação de empresas para execução de serviços. Permitindo que a organização contratante concentre-se no produto final, ou seja, nos serviços que a organização é capaz de desempenhar com maior qualidade, competitividade e produtividade (PRADO; CRISTOFOLI, 2012).

A atividade de contratação tornou-se uma ferramenta de grande importância, pois adiciona valor ao produto desenvolvido, melhorando os processos internos e os serviços prestados. Dessa forma a terceirização deixou de ser uma intervenção visando exclusivamente a redução de custos e passou a ser uma parte integrante da estratégia da organização (PRADO; CRISTOFOLI, 2012).

A terceirização da Tecnologia da Informação (TI) permite às organizações um acesso simples, rápido e barato a diversas tecnologias. Apesar das vantagens da terceirização de serviços, existem alguns problemas, como por exemplo o relacionamento com os fornecedores, as expectativas sem alinhamento estratégico, os prazos, os custos acima do esperado e a baixa qualidade do produto entregue (PRADO; CRISTOFOLI, 2012).

Este trabalho está inserido no contexto de um órgão público federal, que realiza a terceirização de serviços. Inicialmente fez parte do projeto *Framework de Soluções de TI*, que teve origem por meio de um Termo de Cooperação firmado entre a Universidade de Brasília (UnB) e um órgão da APF, denominado, neste trabalho como *Ministério X*. Esse termo de cooperação esteve vigente entre dezembro de 2012 a agosto de 2015.

O Projeto possui como propósito construir conhecimentos que pudessem tornar viáveis a elaboração e estabelecimento de um *Framework de Soluções de TI* para a Coordenação Geral de TI (CGTI) do *Ministério X*. Esse Projeto foi dividido em três subprojetos, que foram denominados *frentes de trabalho* que são: Gerenciamento de Serviços de TI; Arquitetura de Software; e Gerenciamento de TI, que por sua vez



apoiava atividades como Contratações de Serviços de TI e Processo de Desenvolvimento e Manutenção de Software.

O *Ministério X* solicitou que essas frentes de trabalho incluíssem a manutenção dos softwares já existentes (softwares legados).

Softwares legados são softwares utilizados por um grande período de tempo, eles executam tarefas importantes dentro da organização em que estão inseridos. Estes softwares precisam acompanhar as diversas transformações em que a organização é submetida por meio de manutenções de software (RAMOS, 2011).

A manutenção de software é o total de atividades necessárias para promover uma modificação nos sistemas mantendo suas funções primárias intactas. Ela se inicia quando o software está em funcionamento e apresenta algum problema, necessidade de melhoria ou adaptação (BRUSAMOLIN, 2004; HUNT, 2008).

Existem quatro tipos diferentes de manutenção. A manutenção corretiva que é voltada para corrigir falhas no software, adaptativa que ajusta a utilização do software na organização, evolutiva que agrega novos requisitos, e preventiva que visa reduzir a probabilidade de falhas (ISO, 2006).

De acordo com Bakota et al. (2014) e Sned (2004), qualquer manutenção realizada no software, seja perfectiva, corretiva ou adaptativa, sem o objetivo de melhorar a qualidade do código, acarreta numa modificação no código original. Essa modificação diminui a qualidade interna do mesmo e ao longo do tempo resulta em um aumento do número de manutenções.

Quando os sistemas legados possuem alta demanda por manutenção e as causas dessas demandas não são identificadas e solucionadas, aumenta-se o número de novas demandas, o que também acarreta alto custo para a organização (WARD; BENNETT, 1995; BAOTA et al., 2012).

## 1.2. PROBLEMA

De acordo com estimativas, o total de recursos gastos para realizar a atividade de manutenção em um software representa cerca de 40% a 60% do custo total de um projeto de software (BRUSAMOLIN, 2004). Já para Pressman (2011), a manutenção de software pode ser responsável por mais de 60% de todo o esforço despendido por uma organização. À medida que novas funcionalidades do software surgem, essa porcentagem tende a crescer (PRESSMAN, 2011).

A frequência com que um software apresenta necessidades de manutenções pode indicar que ele possui uma baixa qualidade interna. Quando as causas dessas frequências não são mapeadas, registradas ou solucionadas, as mesmas geram um elevado esforço e custo com manutenção (BAKOTA et al., 2012).

Atualmente, o *Ministério X* passa por uma transição contratual de fornecedores de fábrica de software. Ainda serão definidos quais os sistemas estarão no novo contrato de manutenção de software. Para apoiar essa definição, este trabalho visa trazer uma visão aprofundada de aspectos dos sistemas legados, pois atualmente a organização não possui essa visão de seus sistemas. Todas as conclusões são tiradas de aspectos subjetivos, o Ministério X não possui dados reais que caracterizem esses sistemas de forma objetiva.

Os aspectos subjetivos são retirados através da percepção dos usuários que apresentam queixas relacionadas à falta de qualidade dos sistemas, ao grande número de erros representados em várias requisições de manutenção corretiva e à falta de clareza sobre as documentações dos sistemas e quanto o conteúdo dessas documentações tem auxiliado a atividade de manutenção. Não existe nenhum estudo realizado sobre aspectos dos sistemas legados que trazem dados reais que os caracterizem de forma objetiva.

A fim de caracterizar esses aspectos, este trabalho será baseado em três pontos principais: qualidade interna, documentação e requisições de manutenção corretiva. Toda a caracterização realizada são dados que identificam o atual estado dos softwares. Possuindo como propósito caracterizar os sistemas legados para que o gestor de projetos tenha uma melhor visão sobre seus sistemas e assim direcionar os recursos da organização.

Essa visão do gestor é de grande importância para a organização pois a gestão de sistemas legados é um grande desafio. Quando os recursos da organização são alocados para manutenção dos sistemas legados os demais projetos e demandas da área de TI não são realizados. Por isso, a alocação de recursos deve ser realizada da melhor maneira a fim de evitar gastos de recursos desnecessários com os sistemas legados que poderiam ser investidos para a melhoria de serviços de TI da organização.

Dado o problema, a questão de pesquisa deste trabalho é:

*Como avaliar sistemas legados a fim de apoiar uma organização em seus esforços de melhoria dos serviços de TI a partir de três aspectos: qualidade interna do código, documentação e registros de manutenção corretiva?*

### 1.3. OBJETIVO

O objetivo geral deste trabalho é obter um diagnóstico sobre o atual estado dos softwares legados através de três aspectos: a qualidade interna do código, a documentação e os registros de manutenção corretiva. A partir desse diagnóstico alcançar dados que possam apoiar a organização em seus esforços de melhoria de serviços de TI.

Entre os objetivos específicos deste trabalho estão:

- Obter métricas de software que respondam às necessidades de informação que o *Ministério X* necessita obter sobre o seu legado de softwares;
- Através de métricas de softwares realizar análises sobre a completude da documentação, a qualidade interna do código e mapear os registros de manutenção corretiva dos softwares legados;
- Obter possíveis correlações entre qualidade interna do código, documentação e número de registros de manutenção corretiva;
- Obter indicadores de reconstrução, descarte, melhorar ou manter o sistema.

### 1.4. METODOLOGIA

A metodologia de pesquisa deste trabalho foi classificada quanto à natureza, abordagem, objetivos, procedimentos técnicos e técnicas de coleta de dados.

A natureza da pesquisa foi classificada como aplicada, pois tem como objetivo a geração de conhecimentos para a solução de problemas específicos envolvendo verdades e interesses locais da organização em que o trabalho está inserido (GIL, 2008; MORESI, 2003).

Quanto à abordagem, o estudo foi misto, envolvendo tanto o método qualitativo quanto o método quantitativo. As pesquisas classificadas como qualitativas não requerem o uso de métodos e técnicas estatísticas. O ambiente natural é a coleta direta de dados e o pesquisador é o elemento chave, analisando seus dados de forma indutiva. O pesquisador busca compreender os fenômenos observando, interpretando e descrevendo. Já as pesquisas quantitativas consideram que tudo

pode ser quantificado, traduzindo os dados em números ( MORESI, 2003; MELLO, 2012).

A abordagem quantitativa foi aplicada à medida que se buscava quantificar a documentação dos sistemas, seu tamanho, quantidade de manutenções corretivas e qualidade estática do código. Por outro lado, a pesquisa qualitativa foi empregada ao se investigar o contexto específico em que as manutenções corretivas são realizadas.

Em relação ao objetivo, a presente pesquisa foi caracterizada como descritiva, pois a pesquisa expõe características de determinado fenômeno servindo como base para uma futura explicação (MORESI, 2003). Nesse tipo de pesquisa os pesquisadores analisam seus dados de forma indutiva, tendo como foco o processo e seu significado (GIL, 2008).

Os procedimentos técnicos utilizados neste trabalho como meios de investigação, foram estudo de caso, pesquisa bibliográfica e documental (GIL, 2008; MORESI, 2003).

- **Estudo de Caso:** É um método específico de pesquisa de campo. Realiza investigações de fenômenos à medida que ocorrem sem interferência significativa do pesquisador. Tem como objetivo principal compreender o evento em estudo e ao mesmo tempo desenvolver teorias sobre os aspectos dos fenômenos estudados (GIL, 2008; MORESI, 2003).
- **Pesquisa bibliográfica:** É o estudo desenvolvido com base em materiais publicados em livros, revistas, jornais, redes eletrônicas, ou seja qualquer material acessível ao público em geral. Não é recomendado a utilização de páginas da internet. Essa pesquisa fornece instrumento analítico para qualquer outro tipo de pesquisa (MORESI, 2003). Neste trabalho foram utilizados publicações de bases científicas para apoiar a construção do referencial teórico.
- **Documental:** O desenvolvimento da pesquisa documental segue os mesmos passos da pesquisa bibliográfica mas as fontes de pesquisa são muito mais diversificadas e dispersas. Existem documentos que não recebem nenhum tratamento analítico, como documentos de arquivos de órgãos públicos e instituições privadas, como associações científicas,

igrejas, sindicatos, entre outros. Podem ser incluídos documentos como cartas pessoais, diários, regulamentos, gravações e memorandos (GIL, 2008).

As técnicas de coleta de dados selecionadas para este trabalho foram entrevistas informais e análise documental. A seguir são apresentadas de forma detalhada as técnicas selecionadas:

- **Documentos:** Consiste no estudo de documentos publicados por órgão públicos (GIL, 2008), além da documentação de sistemas e de documentos presentes em ferramentas;
- **Entrevista Informal:** Em entrevistas informais o entrevistador tem uma maior liberdade para desenvolver cada situação ou assunto em qualquer direção que considere adequada. As perguntas realizadas durante a entrevista são abertas e podem ser respondidas dentro de uma conversa informal (MARCONI; LAKATOS, 2003).

Na Figura 1 é apresentada a seleção metodológica utilizada neste trabalho.



Figura 1: Seleção metodológica. Fonte autor.

#### 1.4.1 Estudo de Caso

O estudo de caso é delimitado como um tipo de pesquisa que investiga um acontecimento atual inserido dentro do seu contexto real, onde os limites entre o acontecimento e o contexto não são claramente definidos (YIN, 2001; GIL, 2008).

Realiza uma investigação detalhada de uma ou mais organizações, ou até mesmo de grupos dentro de uma organização com o objetivo de promover uma análise do contexto e dos processos envolvidos no fenômeno em estudo (MORESI, 2003).

A principal preocupação do estudo de caso é a interação entre os fatores e eventos. Sendo um método mais específico de estudo de campo investigando fenômenos a medida com que eles ocorrem, sem qualquer interferência significativa do pesquisador. O foco principal é compreender o evento em estudo e ao mesmo tempo desenvolver teorias mais genéricas a respeito dos aspectos característicos do fenômeno observado. O interesse do pesquisador é a relação entre o fenômeno e seu contexto por isso o fenômeno não pode estar isolado de seu contexto (MORESI, 2003).

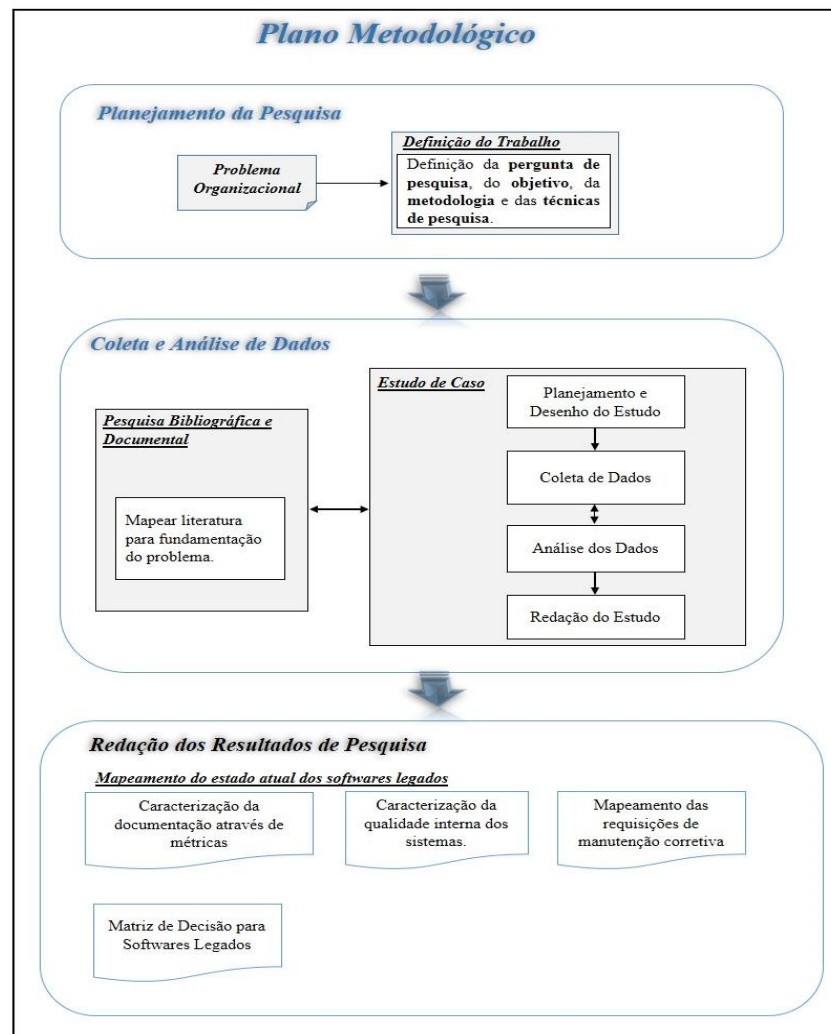
Existem algumas características básicas do estudo de caso. Sendo elas (MORESI, 2003):

- O fenômeno deve ser observado em seu ambiente natural;
- Os dados podem ser coletados por diversos meios;
- Uma ou mais entidades (pessoas, grupo, organização) são examinadas;
- A complexidade da unidade é estudada intensamente;
- As pesquisas realizadas são dirigidas por estágios de exploração, classificação e desenvolvimento de hipóteses do processo de construção do conhecimento;
- Não são utilizados controles experimentais e manipulações;
- Não é necessário a especificação do conjunto de variáveis dependentes e independentes;
- Os resultados dependem fortemente do poder de integração do pesquisador;
- A medida que o pesquisador desenvolve novas hipóteses podem ser realizadas mudanças na seleção do caso ou dos métodos de coleta de dados;
- As pesquisas envolvem questões “como” e “porque” ao invés de frequência ou incidências;
- Possui foco em eventos contemporâneos.

### 1.4.2 Fases do Desenvolvimento

Após a seleção metodológica, tendo como foco principal o estudo de caso, as fases e etapas do trabalho para a execução do trabalho foram estabelecidas com a utilização do plano metodológico.

O plano metodológico compreende as fases de *Planejamento da Pesquisa*, *Coleta e Análise dos Dados* e *Redação dos resultados*. Essas fases são apresentadas na Figura 2.



**Figura 2:** Plano metodológico. Fonte: Autor.

As etapas para a realização do plano metodológico são:

- **Planejamento da Pesquisa:** é a fase que identifica o problema e define a questão de pesquisa, assim como objetivos, geral e específico deste trabalho.

- **Coleta e Análise de Dados:** é a fase onde são definidos e executados dos procedimentos e técnicas utilizados como:
  - Pesquisa Bibliográfica e Documental;
  - Estudo de Caso, com a execução das seguintes etapas: *Planejamento e Desenho do Estudo; Coleta de dados; Análise de Dados; e Redação do Estudo.*
- **Redação dos Resultados de Pesquisa:** é a fase que se executa os procedimentos de Coleta de Dados e da Análise de Resultados.

## 1.5. ORGANIZAÇÃO DO TRABALHO

Este trabalho está organizado em cinco Capítulos. Neste *Capítulo 1 – Introdução* são apresentados: o contexto do trabalho, o problema, os objetivos e a metodologia.

No *Capítulo 2 – Manutenção de Softwares Legados* apresentam-se as informações principais sobre softwares legados e manutenção em diversos aspectos. Esse capítulo apresenta conceitos derivados de manutenção, reengenharia e engenharia reversa.

No *Capítulo 3 – Medição e Métricas* são apresentados os conceitos relacionados às métricas de software e aos métodos de coleta dessas métricas.

No *Capítulo 4 – Materiais e Métodos*, é um detalhamento das fases de desenvolvimento da pesquisa através do estudo de caso.

No *Capítulo 5 – Avaliação dos sistemas legados*, apresentam-se o resultado das métricas e avaliação dos resultados.

No *Capítulo 6 – Análise e comparação dos dados*, é apresentado a relação entre os dados obtidos e matriz de decisão para softwares legados.

No *Capítulo 7 – Conclusão e trabalhos futuros*, é apresentado a conclusão deste trabalho e possíveis trabalhos futuros.



## 2. MANUTENÇÃO DE SOFTWARES LEGADOS

Dado o objetivo definido para este trabalho, neste Capítulo apresenta-se uma contextualização sobre manutenção de software. Inicialmente apresentam-se características relacionadas à manutenção, em seguida apresentam-se conceitos relacionados a softwares legados, reengenharia e engenharia reversa.

### 2.1. MANUTENÇÃO

A manutenção de software pode ser definida como o processo de modificação de um sistema de software que está em operação, deixando suas funções primárias intactas (HUNT, 2008). Essa atividade é iniciada quando a etapa de desenvolvimento do software acaba e a entrega do software é efetuada, ou seja, quando o software já está em funcionamento e apresenta algum problema, necessidade de melhoria ou adaptação (BRUSAMOLIN, 2004).

A atividade de manutenção de software possui as seguintes etapas: entendimento, modificação e revalidação do software. Essas etapas podem consumir uma quantidade de recursos igual ou superior a atividade de desenvolvimento do software. De acordo com estimativas, o total de recurso consumidos representa cerca de 40% a 60% do custo total de um projeto de software (BRUSAMOLIN, 2004).

Dessa forma um dos grandes problemas relacionados à manutenção de software está diretamente ligado a seu custo porém existem outros problemas associados a esta atividade, como a perda ou adiamento das atividades de desenvolvimento, causados pelo fato de que os recursos disponíveis para essa atividade são canalizados na atividade de manutenção, a insatisfação do cliente quando os pedidos de manutenção não são respondidos em tempo hábil; a redução da qualidade do sistema devido à introdução de erros através das alterações no código fonte e a grande diminuição da produtividade dos programadores (PRESSMAN, 2011; BRUSAMOLIN, 2004).

Em softwares legados, as manutenções são geradas de diversas fontes, como o elevado número de solicitações de manutenção por mês, a urgência de pedidos de manutenção (manutenções urgentes podem ter um maior custo por não estarem dentro do planejado), a dificuldade inerente do pedido de manutenção (pedidos de

manutenção podem ter o objetivo de mudar algo em uma parte muito conhecida do sistema ou exigir uma reestruturação total do mesmo) e a dificuldade de compreensão e/ou de modificar o sistema (depende da qualidade da documentação, modularidade do sistema, linguagem de programação utilizada, modelo de dados, entre outros aspectos) (RAMOS, 2004).

## 2.2. TIPOS DE MANUTENÇÃO

A manutenção do software pode ser definida como o processo de corrigir, evoluir ou adaptar programas existentes para que o desempenho do software possa atender às necessidades da organização (RASHID, 2009). Existem diferentes tipos de categorias de manutenção (ISO, 2006), sendo elas:

- **Manutenção Corretiva:** É a modificação de um produto de software após a sua entrega para corrigir falhas. Essas falhas são caracterizadas como defeitos do software e os erros encontrados são nomeados como *bugs*;
- **Manutenção Adaptativa:** É a modificação de um produto de software após sua entrega visando ajustar a utilização do produto de software a um novo contexto no ambiente em que ele está inserido. Essa modificação de ambiente pode ser do *hardware*, regras de negócio, entre outros aspectos que impactam diretamente no software;
- **Manutenção Evolutiva:** Compreende a alteração das funcionalidades do software que já existem para otimizar ou agregar novos requisitos;
- **Manutenção Preventiva:** Sua intenção primária é reduzir a probabilidade de falha de uma máquina ou equipamento, ou ainda a degradação do serviço prestado. Pode ser realizada para adaptar o código visando torná-lo mais claro e facilitar futuras manutenções.

## 2.3. MANUTENIBILIDADE

A manutenibilidade é uma característica de qualidade de software. Possui como foco principal quantificar o grau de facilidade em que um software pode ser corrigido ou aperfeiçoado. Mede o quanto um sistema de software ou um componente é fácil de ser modificado. Dessa forma, um software com um alto grau de manutenibilidade

demandam menos tempo para ser modificado, ou seja, as manutenções são mais fáceis de serem realizadas (BRUSAMOLIN, 2004).

Alguns atributos do software podem impactar diretamente na manutenibilidade do mesmo. Esses atributos são a arquitetura, tecnologia utilizada, documentação do software e compreensibilidade do programa (BRUSAMOLIN, 2004).

A arquitetura do software influencia mais na manutenibilidade do que o próprio algoritmo. A divisão da aplicação em componentes arquiteturais e desses componentes em classes, torna cada unidade do código bem menor do que em uma abordagem monolítica, dessa forma diminui o esforço para modificar o código e o potencial de inclusão de novos erros em uma atividade de manutenção (BRUSAMOLIN, 2004).

Outro fator importante é a tecnologia utilizada durante o desenvolvimento do software. É necessário avaliar se essa tecnologia permite um bom grau de manutenibilidade. A escolha de uma linguagem de programação que permita modificações mais localizadas aumenta a manutenibilidade do software. Modificações mais localizadas levam à menor degradação do código, a qual ocorre em função do número de linhas acrescidas ou modificadas (BRUSAMOLIN, 2004).

Já a documentação do sistema é outro fator de grande importância pois quanto maior o grau de documentação do software menor o tempo que o responsável pela manutenção precisa investir para compreender o mesmo antes de modificá-lo.

Existem estimativas que relatam que programadores gastam entre 47% e 62% do tempo de trabalho tentando compreender o código fonte do software. Dessa forma, quanto maior a compreensibilidade do código, mais manutenível ele se torna (BRUSAMOLIN, 2004).

Para que um software tenha um alto grau de manutenibilidade existem algumas práticas utilizadas durante o desenvolvimento, sendo elas (PIGOSKY, 1996; BRUSAMOLIN, 2004):

- Revisões de averiguação;
- Caminhamentos estruturados;
- Uso de *design* orientado a objetos;
- Assegurar que a cada linha de código tenha no máximo uma declaração;
- Assegurar que os comentários tenham informações úteis;
- Empregar convenções de programação;

- Usar definições de dados comuns;
- Estabelecer padrões para o desenvolvimento de procedimentos e documentos do sistema;
- Estimular a simplicidade;
- Estudar possíveis mudanças futuras e aperfeiçoamentos;
- Medir a complexidade dos componentes do sistema;
- Registrar os pontos fracos do sistema e pontos problemáticos;
- Estabelecer critérios de aceitação para avaliar a qualidade do software do sistema com foco na manutenibilidade;
- Previsão de falhas.

Caso seja realizado uma manutenção ou desenvolvimento de um software violando algumas dessas boas práticas, como consequência poderá acarretar a diminuição da manutenibilidade do mesmo. Essas boas práticas são também uma forma de acompanhar a manutenibilidade de um sistema durante o desenvolvimento (PIGOVISKI, 1996).

Segundo Bakota et al. (2014) existem duas maneiras principais de quantificar a manutenibilidade de um software. A primeira é por meio de análise estática, como uma forma de calcular utilizando métricas, como por exemplo: complexidade ciclomática, proporção de erros de codificação graves, duplicação de código, e entre outras. A segunda representa atributos de qualidade de sentido subjetivo que não podem ser calculados diretamente a partir do código fonte, como quantidade de manutenções realizadas e estabilidade do código.

#### 2.4. MANUTENÇÃO E CUSTO

Os custos com manutenção são iniciados depois da entrega do software. Antes dessa entrega, todos os custos são considerados custos de desenvolvimento. Os custos de desenvolvimento são os custos do projeto e devem ser cobrados dentro do orçamento do projeto, já os custos de manutenção são referentes a todas as modificações feitas no software após a sua entrega (SNEED,2004).

O número de manutenções realizadas em um sistema de software, sem custo adicional ao custo do desenvolvimento, representam 15% do custo total do software. Esse percentual só é válido dentro do período de garantia do software, normalmente se refere somente às correções essenciais e adaptações necessárias, somente para

manter o sistema em funcionamento. As otimizações necessárias, eliminação de falhas, melhorias, adaptações e correções fora do período de garantia são cobradas e acrescidas a esse percentual (SNEED,2004).

As manutenções realizadas em um software danifica sua estrutura original tornando necessário novas manutenções, o que gera um crescente custo para a organização. À medida que o tempo passa, o custo para implementar uma mudança será maior. Isso ocorre para todos os sistemas de software, independente da sua aplicação de domínio ou base tecnológica (JACOBSON, 1991; SNEED,2004).

Como os sistemas legados muitas das vezes não estão no período de garantia do software e já sofreram diversas modificações, eles geram um elevado e crescente custo com suas manutenções para a organização. Sendo necessário prever esse custo e buscar alternativas para a diminuição do mesmo (WARD; BENNETT, 1995, RAMOS, 2011).

Existe uma relação direta entre os custos com manutenção desses sistemas e o quanto ele é manutenível ou não. Dessa maneira uma forma de prever futuros custos com manutenção é avaliar a manutenibilidade do sistema. Caso o software possua um custo de desenvolvimento baixo, mas não é manutenível, a organização certamente arcará com futuros gastos, pois seus custos com manutenção serão elevados (BAKOTA et al., 2012).

Outros quatro fatores que podem prever futuros custos com manutenção de software são: análise estática do código, análise dinâmica, análise de defeitos e análise de produtividade. Para realizar essa previsão de gastos esses fatores precisam ser medidos (SNEED, 2004).

A análise estática de software tem como resultado métricas de código, como tamanho, complexidade, duplicação, e entre outros. A análise dinâmica tem como objetivo final encontrar vulnerabilidade do código em tempo de execução da aplicação. A análise de defeitos classifica o número e os tipos de defeitos. Os erros reais são contabilizados e a densidade de defeitos é calculada. A análise da produtividade reúne dados como relatórios de erros, pedidos de mudança, desenvolvimento de reengenharia, e entre outros aspectos (SNEED, 2004).

## 2.5. SOFTWARES LEGADOS

Softwares legados são sistemas que estão na organização por um grande período de tempo, por isso ele incorpora anos de experiência e refinamento, pois as informações do negócio e os procedimentos que estão embutidos podem não estar documentados. Em contrapartida eles possuem todos os vícios e defeitos vigentes à época do seu desenvolvimento além de novos defeitos inseridos nas modificações já realizadas (WARD; BENNETT, 1995).

Como as regras de negócio de um organização mudam constantemente, o sistema legado precisa sofrer modificações para que continue atendendo às necessidades da organização (RAMOS, 2004). Essas modificações geram desafios técnicos, como o de verificar se existem efeitos colaterais após uma alteração; e não técnicos, como estimar o custo de uma alteração (WARD; BENNETT, 1995).

Bennet (1991) identifica problemas críticos relacionados aos softwares legados, como:

- **Desestruturação e falta de entendimento do código:** Esse fator se deve principalmente porque o código não foi desenvolvido utilizando os métodos da programação estruturada;
- **Programadores que não participaram do desenvolvimento do sistema:** A falta de participação na construção de um produto de software faz com que os programadores sintam dificuldade em entender e mapear as funcionalidade do código fonte;
- **Documentação desatualizada:** As modificações ocorridas no software durante seu funcionamento não produzem uma atualização na documentação existente. Dessa forma a documentação não auxilia a equipe de manutenção;
- **Dificuldade de prever:** Os efeitos colaterais de uma nova modificação são difíceis de serem previstos;
- **Dificuldade de administrar as diversas alterações concorrentes.**

Todo sistema de software, principalmente os softwares legados, possuem um tempo de vida limitado, e cada alteração efetuada degrada a sua estrutura, fazendo com que novas manutenções se tornem cada vez mais difíceis. Pois essas sucessivas manutenções geram degradações no software aumentando a dificuldade

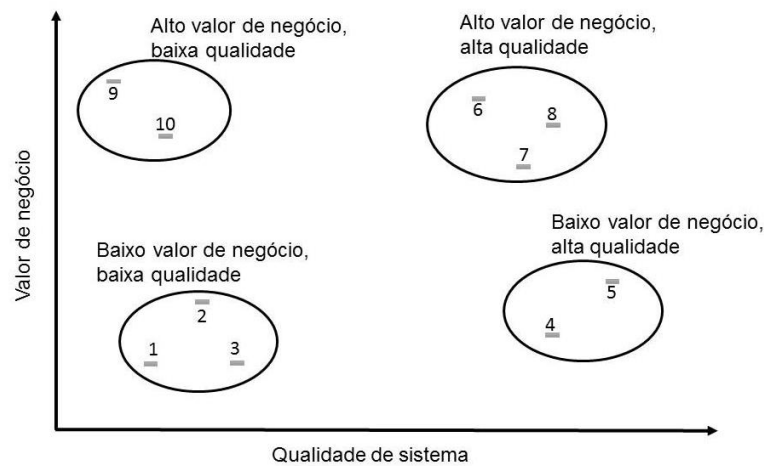
de entendimento do mesmo, além de diminuir sua qualidade interna tornando as manutenções seguintes mais caras (BAKOTA et al., 2012).

Como manutenções em sistemas legados envolvem um alto custo e as organizações possuem uma restrição de orçamento destinado para a manutenção, elas precisam decidir sobre como obter o melhor retorno de seu investimento. Essa decisão é realizada por meio de uma avaliação dos seus sistemas legados. Essa avaliação deve permitir traçar uma estratégia mais adequada para seus sistemas. Existem quatro opções estratégicas que podem ser tomadas, sendo elas (SOMERVILLE, 2011):

- **Descartar completamente o sistema:** Essa opção deve ser escolhida quando o sistema não está mais contribuindo efetivamente para o processo dos negócios. Isso ocorre quando os processos de negócio se alteram desde que o sistema foi instalado e já não dependem dos sistemas legados;
- **Deixar o sistema inalterado e continuar com as manutenções regulares:** Essa opção deve ser escolhida quando o sistema ainda é necessário, é bastante estável e os usuários do sistema fazem poucas solicitações de mudança;
- **Reestruturar o sistema para melhorar a sua manutenibilidade:** Essa opção deve ser escolhida quando a qualidade do sistema foi degradada pelas mudanças, e novas mudanças para o sistema ainda estão sendo propostas;
- **Substituir a totalidade ou parte do sistema por um novo sistema:** Essa opção deve ser escolhida quando fatores como *hardwares* novos não permitem que o sistema antigo possa continuar em operação. Em muitos casos, uma estratégia de substituição evolutiva pode ser adotada, na qual sempre que possível, os componentes principais do sistema são substituídos por sistemas de prateleira com outros componentes reusados.

A avaliação de um sistema legado deve ser realizada olhando por uma perspectiva de negócio com um olhar mais técnico. A perspectiva de negócio deve ser feita para analisar se o negócio realmente necessita do sistema. A perspectiva técnica permite avaliar a qualidade do software de aplicação, de apoio de software e de *hardware*. A organização precisa sempre realizar uma combinação do valor de

negócio e da qualidade do sistema para obter uma decisão do que deve ser feito com um sistema legado (SOMERVILLE, 2011).



**Figura 3:** Avaliação de sistemas legados. Fonte:(SOMERVILLE, 2011)

Dessa forma, existe alguns critérios para avaliar ou classificar possíveis decisões a serem tomadas sobre o futuro dos softwares legados presentes na organização. Esses critérios estão especificados a seguir (SOMERVILLE, 2011):

- **Baixa qualidade e baixo valor de negócio:** Manter esses sistemas em funcionamento implica em um alto custo e um baixo retorno para o negócio. Esse sistemas devem ser descartados;
- **Baixa qualidade, alto valor para o negócio:** Esses sistemas proporcionam uma contribuição importante para o negócio, dessa forma eles não podem ser descartados. Contudo, sua baixa qualidade significa que seu custo de manutenção é alto. Esses sistemas devem ser reestruturados visando o aumento de sua qualidade;
- **Alta qualidade e baixo valor de negócio:** Esses sistemas não contribuem muito para o negócio, mas seus custos de manutenção podem não ser altos. As manutenções desses sistemas devem ser mantidas. Caso as mudanças necessárias fiquem caras o software deve ser descartado;
- **Alta qualidade e alto valor para o negócio:** Esses sistemas precisam ser mantidos em operação. A alta qualidade indica que não há



necessidade de investimento na transformação ou substituição do sistema.

Para realizar uma avaliação com o objetivo de identificar o valor de negócio dos sistemas legados, é necessário realizar o levantamento de algumas informações juntamente aos responsáveis pela área de negócio da organização. A investigação deve conter os seguintes dados (SOMERVILLE, 2011):

- **O uso do sistema:** Um sistema usado ocasionalmente ou por um pequeno número de pessoas pode ter um valor de negócio baixo. Um sistema legado por ter sido produzido para necessidades de negócio que tenham se alterado ou que agora são atendidas de forma mais eficaz;
- **Os processos de negócio em que são apoiados:** Quando um sistema é produzido, processos de negócio são desenvolvidos para explorar a capacidade dele. Com a mudança de ambiente os processos de negócio podem se tornar obsoletos. Dessa forma, sistemas podem ter um valor de negócio baixo por obrigar o uso de processos de negócios ineficientes;
- **Confiança no sistema:** A confiança no sistema não é apenas um problema técnico, mas também um problema de negócio. Se o sistema não for confiável e os problemas afetarem diretamente os clientes do negócio, significa que os profissionais serão desviados de outras tarefas para resolverem esses problemas. O sistema terá um valor de negócio baixo;
- **As saídas do sistema:** É necessário avaliar se as saídas do sistema interferem no bom funcionamento do negócio. Se o negócio depender dessas saídas o sistema terá um alto valor de negócio, mas se o sistema produzir saídas que raramente forem usadas, ele terá um baixo valor de negócio.

## 2.6. ENGENHARIA REVERSA

O processo tradicional de produção de um sistema de software caracterizado pelas atividades progressivas do ciclo de vida, como análise de requisitos, desenvolvimento do projeto, codificação e testes, são atividades que partem de um alto nível de abstração para um baixo nível de abstração. Essa fase é conhecida como engenharia progressiva (CHIKOFSKY, 1990).

O processo inverso à engenharia progressiva é caracterizado pelas atividades retroativas do ciclo de vida, como a documentação de um sistema depois do encerramento da sua fase de construção. São atividades que partem de um baixo nível de abstração para um alto nível de abstração. Esse processo é conhecido como engenharia reversa (CHIKOFSKY, 1990).

A engenharia reversa é um processo de exame e compreensão do software existente, com o objetivo de recapturar ou recriar o projeto decifrando os requisitos que já foram implementados pela atividade de desenvolvimento, apresentando-os em um grau ou nível mais alto de abstração. Não é a criação de um novo software (CHIKOFSKY, 1990).

A técnica de engenharia reversa tem por objetivo principal contribuir com o entendimento e com uma futura modificação e revalidação do software, aumentando assim a manutenibilidade do mesmo. Isso é feito por meio de um processo de análise, que procura criar representações dos softwares em uma forma mais clara ou em um nível mais alto de abstração que o código fonte (CHIKOFSKY, 1990).

Existem duas categorias de engenharia reversa. Essas categorias são obtidas a partir do nível de entendimento do sistema e o escopo das informações utilizada. Essas duas categorias são: visualização do código e entendimento do programa (CHIKOFSKY, 1990).

A visualização do código, também denominada redocumentação, é a criação de representações a partir de informações obtidas apenas da análise do código fonte. Tem como intenção recuperar a documentação que já existiu ou que deveria ter existido sobre o sistema. Já o entendimento do programa, também denominado de recuperação de projeto, é o conhecimento do domínio das informações externas, e as deduções são adicionadas às observações feitas sobre o sistema por meio de exame do mesmo, possuindo como objetivo obter informações com nível mais alto de abstração (CHIKOFSKY, 1990).

Muitas das vezes, o descarte do sistema atual e redesenvolvimento de um novo sistema não é viável, pois o sistema que já está em funcionamento possui um acúmulo de experiências e informações obtidos durante seu tempo em funcionamento. Pode ser economicamente inviável descartar o alto investimento financeiro já atribuído ao software (BENNET, 1995).

Dessa forma, uma solução viável, que ao longo do tempo seja capaz de reduzir o esforço com manutenções, é através da engenharia reversa. A engenharia reversa

é capaz de produzir atributos do software que tornam o esforço com manutenção menor (BENNET, 1995). A engenharia reversa é parte de manutenção que auxilia na compreensão do sistema para a realização de manutenções necessárias (CHIKOFSKY, 1990).

## 2.7. REENGENHARIA

A ideia básica da reengenharia é extrair do código fonte as informações de projeto e especificações do sistema de forma que elas sejam reformuladas e reconstruídas, resultando em um software mais fácil de ser mantido (BANNETT, 1991).

Reengenharia de software consiste na reimplementação de um sistema em uma forma mais fácil de ser mantida. Podendo redocumentar o sistema ou realizar uma reestruturação. Os programas podem ser traduzidos para uma linguagem de programação mais moderna e implementados em uma plataforma mais distribuída. A reengenharia tem como objetivo fazer sistemas de software mais flexíveis e fáceis de modificar, de forma a atender as necessidades de negócio da organização (BANNETT, 1991).

A reengenharia envolve primeiramente um processo de engenharia reversa para a recuperação de informações importantes sobre o sistema e posteriormente a reconstrução do mesmo (engenharia progressiva). Os métodos para a realização da engenharia reversa devem proporcionar uma maior qualidade do sistema (BANNETT, 1991).

A reengenharia de software também pode ser descrita como a reorganização e modificação de sistemas de software existentes, parcial ou totalmente, com o objetivo final de torná-los manuteníveis (SOMMERVILLE, 2011).

Existe uma grande diferença entre o desenvolvimento de um novo software e reengenharia de software. Essa diferença está relacionada ao ponto de partida de cada uma das atividades. A atividade de desenvolvimento de um novo software tem como base a engenharia progressiva, que se inicia com uma especificação escrita do software que será construído, enquanto que a reengenharia inicia-se tomando como base um sistema já desenvolvido (BANNETT, 1991).

Encontramos também uma diferença entre os objetivos da reengenharia e da engenharia reversa. O objetivo da engenharia reversa é derivar o projeto ou

especificações de um sistema a partir do seu código fonte (SOMMERVILLE, 2011). O objetivo da reengenharia é produzir um sistema novo e mais manutenível. A engenharia reversa é usada como parte desse processo de reengenharia, para o entendimento do sistema a ser construído (BANNETT, 1991).

### 2.7.1 Processo de Reengenharia

O processo de reengenharia pode ser definido a partir da junção da engenharia reversa com a engenharia progressiva, de alterações de funcionalidades ou de alteração de implementação. A alteração de funcionalidade ocorre devido a mudanças na área de negócio da organização ou das necessidades do usuário. A alteração de implementação altera a forma de implementar o sistema, ou seja, ocorrem devido às alterações no ambiente de operação do software e ou na linguagem de implementação (WARDEN, 1992).

Para a boa execução de um processo de reengenharia deve-se considerar alguns pontos (WARDEN, 1992), sendo eles:

- A reengenharia deve ser executada somente quando seu custo benefício for aceitável para a organização;
- A reengenharia implica em melhoria através de reprojetos;
- Reengenharia deve reconhecer projetos ruins, mas manter projetos bons e simples, mesmo se esses projetos forem desestruturados;
- A fase onde se realiza a engenharia reversa deve ser dirigida por tipos de problemas, os quais necessitam ser identificados;
- Problemas são identificados e expressos como violações às técnicas de projeto estruturado e às regras de programação, ou outras que o usuário definir;
- As ferramentas estabelecidas devem ser adequadas aos processos de reengenharia.

Os pontos que devem ser realizados para executar o processo de reengenharia de um sistema são (JACOBSON, 1991):

- **Engenharia reversa:** É a identificação dos componentes do sistema e como eles se relacionam, de forma a criar uma decisão mais abstrata sobre o sistema;

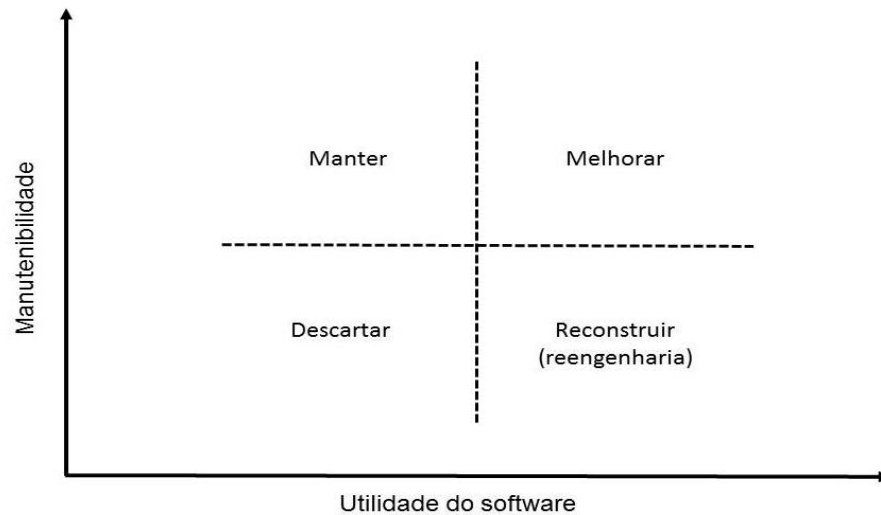
- **Decidir sobre alterações nas funcionalidades:** As alterações de funcionalidades são alterações nos requisitos, que, de acordo com o usuário, devem ser implementadas no sistema. Esse passo é executado utilizando-se as abstrações de mais alto nível, obtida no passo anterior;
- **Reprojetar o sistema:** o reprojeto do sistema parte das abstrações de alto nível, obtidas nos passos anteriores, para uma representação mais concreta, ou seja, é a execução da engenharia progressiva reimplementando o sistema.

### 2.7.2 Reengenharia Aplicada

Para dar início a um processo de reengenharia, levando em consideração a complexidade de gerenciamento do processo, é necessário seguir alguns passos citados a seguir (SNNED, 1995):

- **O projeto deve ser justificado:** Deve ser realizado um estudo sobre os sistemas presentes na organização, seu processo de manutenção e qual a importância do sistema nos processos internos da organização. Essa justificativa deve ter como base o retorno de investimento que a organização terá ao realizar a reengenharia em seus sistemas dando ênfase no aumento da qualidade do software, o aumento da manutenibilidade, a melhora no processo de manutenção e o consequente aumento do valor do software para a organização;
- **O portfólio de projetos deve ser analisado:** Deve estabelecer critérios para a escolha dos sistemas que são candidatos a reengenharia;
- **Deve-se obter uma estimativa de custo:** O custo da realização de uma reengenharia em um sistema deve ser estimado a partir da identificação e avaliação de todos os componentes de software que serão reconstruídos. Esses componentes devem conter a reengenharia do software, testes e reengenharia de dados;
- **Realizar uma análise dos custos benefícios:** Realizar uma comparação entre o custo esperado com manutenções e o custo estimado para realizar a reengenharia do sistema. Se os custos de reengenharia forem altos, pode-se optar por outras soluções, como comprar software pronto ou realizar redesenvolvimento.

Devido ao alto custo para se aplicar a reengenharia em um sistema, deve-se analisar os custos benefícios de se aplicar a reengenharia levando em consideração a viabilidade entre substituir, manter, melhorar ou efetuar a reengenharia nesse sistema (JACOBSON, 1991).



**Figura 4:** Matriz de decisão. (JACOBSON, 1991; adaptado).

Segundo Jacobson (1991), alterabilidade de software é a facilidade com que o software consegue ser modificado. Neste trabalho o conceito de alterabilidade é assumido como manutenibilidade. Por meio da manutenibilidade do software e da utilidade do mesmo para a organização, pode-se aplicar a matriz de decisão apresentada na Figura 4 para nortear as tomadas de decisões, como (JACOBSON, 1991):

- Sistemas com uma alta manutenibilidade, porém com pouca utilidade para a organização, devem ser mantidos, ou seja, devem continuar realizando manutenções normalmente;
- Sistemas com baixa manutenibilidade e pouca utilidade para a organização devem ser descartados ou substituídos por outros;
- Sistemas com alta manutenibilidade e com grande utilidade para a organização devem continuar evoluindo;
- Somente sistemas com baixa manutenibilidade e de alta utilidade para a organização devem ser reconstruídos (reengenharia).

Os custos de um processo de reengenharia estão diretamente ligados a profundidade ou extensão que a reengenharia será executada. Tendo também

influência a qualidade do software que será reconstruído, as ferramentas de suporte disponíveis para auxiliar a reengenharia, a extensão da conversão dos dados requerida e a disponibilidade de pessoas qualificadas para realizar a reengenharia (JACOBSON, 1991).

Em alguns casos, os custos da realização de reengenharia de um sistema são mais altos que o custo de redesenvolvimento do mesmo. No entanto a reengenharia apresenta algumas vantagens sobre o redesenvolvimento de sistemas. Se o sistema é de grande importância para a empresa, a reengenharia apresenta um menor risco do que o redesenvolvimento do mesmo, pois a reengenharia garante que as funções primárias do sistema sejam mantidas, aumentando sua qualidade (JACOBSON, 1991).

Caso a decisão tomada seja a do redesenvolvimento do sistema, a melhor opção não é substituir um sistema antigo por um sistema completamente novo, pois essa mudança requer muitos recursos. Devem ser encontradas maneiras de substituição gradual, substituindo partes dos sistemas legados sem perder completamente os investimentos feitos até o momento. Os sistemas antigos devem ser modernizados de forma gradual e, eventualmente, todo o sistema será substituído (JACOBSON, 1991).

### 3. MEDIÇÃO E MÉTRICAS

Este capítulo aborda conceitos relacionados à medição de software, ao processo de medição de software, às métricas em diversos aspectos e à abordagem GQM.

#### 3.1. MEDIÇÃO DE SOFTWARE

A medição de software é a derivação de um valor numérico para um atributo de um componente de software, sistema ou processo. A medição é capaz de tirar conclusões sobre a qualidade do software, ou avaliar a eficácia dos métodos, das ferramentas e do processo de software (SOMERVILLE, 2011).

O processo de medição de um software pode ser formalmente definido como o processo de atribuir números e símbolos a entidades do mundo real. Essa atribuição é realizada de forma a descrever as entidades de acordo com regras claramente definidas. A medição de software tornou-se prática comum a muitas organizações, sendo um mecanismo de avaliação contínua do projeto de software. Ela é realizada através de métricas de software (RAMOS, 2004).

#### 3.2. MÉTRICAS DE SOFTWARE

Métricas de software são um conjunto de medidas de um processo ou produto de software. O produto de software pode ser medido durante todas as fases de construção até que o sistema seja finalizado, e isso inclui seu código fonte e as documentações produzidas durante o seu desenvolvimento (SOMERVILLE, 2011).

Métricas de software podem ser classificadas como métricas de controle ou métricas de previsão. Métricas de controle dão suporte ao processo de gerenciamento e são associadas aos processos de software. Métricas de previsão ajudam a prever as características do software, e são associadas ao software em si, tais métricas são conhecidas como métricas de produto (SOMERVILLE, 2011).

Métricas de controle são também uma forma de gerenciamento, podem influenciar a tomada de decisões gerenciais. Os gerentes usam métricas de processo para decidir se é necessário realizar alterações nos processos. As métricas de previsão são utilizadas para ajudar a estimar o esforço necessário para realizar alterações no software (SOMERVILLE, 2011).



As métricas podem ser utilizadas para a obtenção de medidas bem definidas e viáveis que produzem como resultado uma avaliação do software durante o seu desenvolvimento. Podem ser utilizadas como uma forma de gestão durante o desenvolvimento ou após o sistema ter sido entregue, pois utilizam a identificação, medição e controle dos parâmetros essenciais do software (FENTON; PFLEEGER, 1998).

Os critérios a seguir devem ser definidos para a escolha de métricas eficazes para o objetivo esperado (FENTON; PFLEEGER, 1998):

- Definir de forma clara qual o objetivo principal de medição;
- O valor da métrica deve estar atrelado ao seu objetivo de medição;
- A métrica deve ser formalmente definida;
- A métrica precisa medir de forma eficaz;
- Pequenas mudanças no software não devem causar grandes variações nas métricas;
- As métricas precisam ser mapeadas de forma a obter o comportamento dos dados analisados através do manuseio dos números obtidos;
- Os resultados das métricas devem ser independentes da linguagem de programação utilizada.

As métricas de software podem ser classificadas quanto ao domínio da aplicação, critério utilizado na sua definição e método de obtenção da medida. A maneira mais vasta de classificação é quanto ao objetivo da métrica. As métricas podem ser classificadas também quanto ao seu método de obtenção, denominadas métricas primárias e métricas compostas. Métricas primárias são aquelas que são obtidas através de uma única métrica, já as métricas compostas são combinações de duas ou mais métricas (FENTON; PFLEEGER, 1998).

Quanto aos critérios utilizados para definir as métricas, elas são classificadas como métricas objetivas e métricas subjetivas. As objetivas são obtidas através da utilização de regras bem definidas, sendo essa a melhor forma de realizar futuras comparações. As subjetivas são dependentes do sujeito que está realizando a medida, dificultando a comparação e uma futura reprodução da medida realizada (FENTON; PFLEEGER, 1998).

As métricas podem possibilitar uma avaliação de software de forma sistemática, eficiente, eficaz e de baixo custo, independente do momento em que são utilizadas.

Desde que os valores obtidos pelas métricas sejam sempre os mesmos, não tendo relação com as condições ou o indivíduo que realiza as medidas (FENTON; PFLEEGER, 1998).

Toda métrica precisa estar associada a uma escala de medições que proporcione significado ao valor no seu cálculo. Elas precisam ser coletadas em um modelo de dados específicos que podem envolver cálculos ou análise estática. É necessário considerar o tipo de informação que se deseja obter através das métricas utilizadas (FENTON; PFLEEGER, 1998).

Existem cinco tipos de dados de medidas que foram reconhecidos por estatísticos para métricas de software, sendo eles (FENTON; PFLEEGER, 1998; RAMOS, 2011):

- **Nominal:** A ordem dos valores não possui significado para a interpretação dos valores de cada atributo;
- **Ordinal:** Os valores obtidos possuem uma ordem, mas a distância entre os valores não possui significado;
- **Intervalo:** A ordem dos resultados e o tamanho dos resultado são importantes, mas as proporções entre eles não são necessariamente válidas;
- **Racional:** A ordem dos resultados e o tamanho dos resultado são importantes, as proporções entre os resultados são válidas;
- **Absoluta:** Realizada através da contagem do número de elementos de uma entidade.

### 3.3. PROCESSO DE MEDIÇÃO DE SOFTWARE

O processo de medição de um software pode ser formalmente definido como o processo de atribuir números e símbolos a entidades do mundo real. Essa atribuição é realizada de forma a descrever as entidades de acordo com regras claramente definidas. A medição de software tornou-se prática comum em muitas organizações sendo um mecanismo de avaliação contínua do projeto de software (RAMOS, 2004).

Para realizar o processo de medição do software é necessário analisar cada componente separadamente, usando um conjunto de métricas. Os valores dessas métricas podem ser comparados com diferentes componentes e talvez com dados históricos de medição, obtidos em projetos anteriores (SOMERVILLE, 2011).

Os estágios principais do processo de medição são (SOMERVILLE, 2011):

- **Escolher medições a serem efetuadas:** As questões que a medição está destinada a responder devem ser formuladas e as medições necessárias para responder a essas questões devem ser definidas;
- **Selecionar componentes a serem avaliados:** Pode ser realizada somente uma seleção de um conjunto representativo de componentes para a medição, que permita fazer uma avaliação global da qualidade de um sistema;
- **Medir características de componentes:** Os componentes selecionados são medidos e os valores das métricas coletadas são computados;
- **Identificar medição anômalas:** Os valores obtidos através das medições devem ser analisados, procurando valores anormalmente altos ou baixos para cada métrica;
- **Analisar os componentes anômalos:** Ao identificar os componentes anômalos, deve-se verificar se esses valores anômalos indicam que a qualidade do sistema está comprometida. Pode haver outra razão para o valor que não está relacionada com a qualidade do componente.

#### 3.4. GQM (*Goal Question Metric*)

O GQM tem como base sustentar a identificação de métricas apropriadas de acordo com o contexto e os objetivos da avaliação realizada, sustentando também a análise, legitimidade dos dados coletados, interpretação e armazenamento (BASILI, 1994).

A ideia principal do GQM é que a medição deve ser o objetivo orientador, sendo uma metodologia que parte do princípio *top-down* com definição de um objetivo em que a medição é realizada. Do objetivo definido são derivadas questões, e, em seguida, cada questão é refinada em métrica, que, quando medida, deve fornecer informações para responder as questões. O resultado das métricas deve ser capaz de indicar se os objetivos estão sendo atendidos (RAMOS, 2004).

Dessa forma o GQM é dividido em três níveis. Sendo eles ( BRASILI, 1994):

- **Nível Conceitual (*Goal*):** Envolve a especificação do objetivo da aplicação do GQM. Para cada objetivo deve existir um único propósito e

um único ponto de vista em relação ao ambiente especificado. Os objetivos podem ser definidos a partir do processo, produto ou recurso;

- **Nível Operacional (*Question*):** A questão representa a necessidade de obter informação utilizando uma linguagem natural. Pode existir uma ou mais questões para cada categoria estabelecida e suas respostas devem estar de acordo com o objetivo especificado. Possui como propósito caracterizar o objetivo de medição;
- **Nível Quantitativo (*Metric*):** A métrica é utilizada para especificar os dados ou as informações que se deseja obter durante as avaliações em termos quantitativos. Para cada questão estabelecida são utilizadas uma ou mais métricas. As métricas podem ser do tipo objetiva, dependendo apenas do objeto e não do ponto de vista (por exemplo: tamanho de um programa, o número de erros, entre outros) ou subjetiva, dependendo tanto do objeto quanto do ponto de vista (por exemplo: nível de satisfação utilizado, clareza de uma especificação, entre outros).

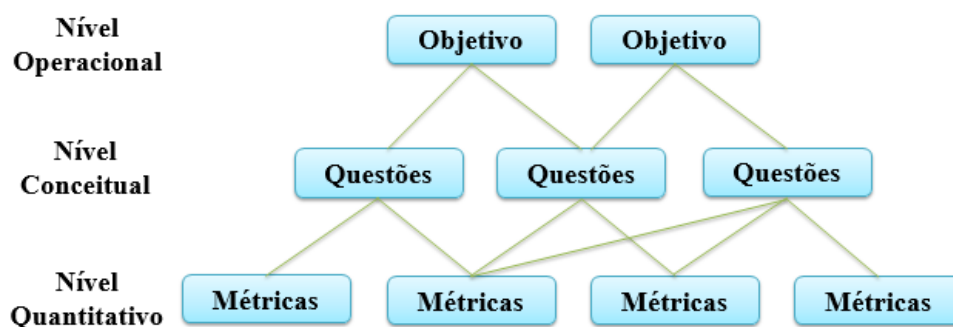


Figura 5: GQM. Fonte: (BASILI, 1994, adaptado)

Para definir um modelo GQM seguimos quatro fases principais: planejamento, definição, escolha dos dados e interpretação. O objetivo principal da fase de planejamento é escolher um item desejado e escolher a melhor área de aplicação, com base nos objetivos definidos pela a organização. Na fase de definição são estabelecidos os objetivos, questões e métricas. A fase de coleta de dados corresponde à definição do procedimento de extração dos dados. E por último a fase de interpretação dos dados. Nessa fase é verificado se os dados obtidos respondem as questões realizadas. Com base nessa resposta é possível concluir se os objetivos de medição foram atendidos ou não (RAMOS, 2004).

O GQM auxilia na identificação de métricas significativas para a análise e interpretação dos dados coletados. Para apresentar um bom resultado, o GQM deve ser planejado e executado seguindo os aspectos a seguir (DIFFERDING, 1996):

- A meta de avaliação deve ser o meio pelo o qual a análise deve ser realizada;
- As medidas devem ser derivadas de uma forma *top-down*, baseadas em objetivos e questões;
- Cada medida deve ter como base um fundamento lógico subjacente, que é explicitamente documentado. O fundamento lógico utilizado deve guiar a análise e interpretação dos dados;
- Os dados que são coletados através das medidas precisam ser interpretados através de um modelo *bottom-up*;

Para um melhor desenvolvimento das medições realizadas através do GQM, podem ser definidos dois parâmetros iniciais para métricas, sendo eles: (i) as métricas devem proporcionar uma coleta rápida; e (ii) quando possível as métricas devem ser automatizadas (RAMOS, 2004).

A vantagem de usar a abordagem do GQM é que o mesmo separa os interesses organizacionais (os objetivos) dos interesses específicos de processos (as questões). O método fornece uma base para decidir quais dados devem ser coletados e sugere que os dados coletados devem ser analisados de diferentes maneiras, dependendo da questão que eles devem responder (SOMERVILLE, 2011).

### 3.5. VERIFICAÇÃO DE SOFTWARE

A verificação de software é uma atividade que determina se os produtos de software atendem aos requisitos ou condições impostas nas atividades anteriores, ou seja, é a confirmação ou o fornecimento de evidências objetivas de que o sistema atende aos requisitos especificados (ISO/IEC, 1998).

### 3.6. INSPEÇÃO DE SOFTWARE

A inspeção de software é a atividade em que o sistema é revisto para se encontrar erros, omissões e anomalias. Geralmente possui foco no código fonte do sistema. Essa atividade utiliza o conhecimento precedente do sistema, domínio de

aplicação e a linguagem de programação. Possui como objetivo descobrir erros (SOMMERVILLE, 2011).

É uma forma de verificação de características de qualidade do código fonte, principalmente em relação a sua conformidade e manutenção. Apesar de inspeções serem focadas no código-fonte, também é possível utilizá-las para avaliar requisitos ou modelos de projeto. Entre as vantagens de se realizar uma inspeção de código, estão (SOMMERVILLE, 2011):

- A inspeção de código é capaz de descobrir grandes quantidades de erros em uma sistema em uma única inspeção realizada;
- Sistemas que ainda não foram concluídos podem ser inspecionados, sendo uma forma de acompanhar os erros presentes no sistema durante seu processo de construção;
- A inspeção garante que quando houver uma interação entre erros, um erro não mascare a presença do outro;
- Uma inspeção realizada no código pode descobrir atributos de qualidade mais amplos como conformidade de padrões e o grau de manutenibilidade de um sistema.

### 3.7. ANÁLISE ESTÁTICA DE CÓDIGO

A análise estática de código é aplicada para a detecção de defeitos. Ela consiste em uma verificação automática do código fonte em tempo de compilação antes mesmo da realização de testes de unidade do software. É capaz de capturar alocações de erros e falhas de segurança. A análise estática tem se tornado um método importante na verificação do código fonte (CHELF, 2009).

O objetivo principal da análise estática é encontrar vulnerabilidade no sistema reduzindo mais de 50% das falhas do software após a sua entrega para a organização. Quando é realizada durante o desenvolvimento é capaz de reduzir seu custo de 20% a 30% (CHELF, 2009).

Ao efetuar uma análise estática de código em um sistema específico, no decorrer do seu tempo de vida útil, ocasiona-se uma diminuição do esforço com manutenções (PLOSCH, 2008).

### 3.8. FERRAMENTAS DE ANÁLISE ESTÁTICA

Ferramentas de análise estática tentam identificar e quantificar a qualidade do código. O objetivo principal é obter o máximo de informações qualitativas e quantitativas, verificando completamente o código fonte, a fim de verificar padrões de “maus cheiros”, erros de programação, violações de regras de programação, entre outros aspectos (PLOSCH, 2008).

Ferramentas de análise estática foram desenvolvidas para diferentes linguagens de programação. Os analisadores estáticos são ferramentas capazes de passar por todo código fonte (SOMMERVILLE, 2011). Algumas ferramentas para análise estática de código estão a seguir (SIEBRA, 2014):

- Checkstyle;
- Clan;
- Coverity;
- CppCheck;
- Findbugs;
- Flawfinder;
- HP Fortify;
- PMD;
- Red Lizard Goanna; e
- SonarQube.

### 3.9. SONARQUBE

O *SonarQube* é uma ferramenta de análise estática de código que possibilita que os relatórios, resultados da análise do código fonte gerados, tenham uma visualização de forma clara e prática. É uma ferramenta gratuita e de código aberto que oferece uma visão de qualidade do código ao longo do projeto. Ele é composto por três componentes (CAMPEBELL, 2014):

- Banco de dados próprio para armazenar as configurações da ferramenta e os resultados das análises de código realizadas;
- Um *web server* que permite a visualização dos resultados das análises através de um *browser*;

- Um ou mais analisadores responsáveis pelas análises dos projetos. Como checkstyle, PMD, entre outros.

O *SonarQube* realiza a análise de sete áreas de qualidade de código, sendo elas (CAMPBELL, 2014):

- Arquitetura e Design;
- Duplicações;
- Testes;
- Complexidade;
- Bugs em potencial;
- Regras de codificação;
- Comentários.

A ferramenta *SonarQube* possui diversas vantagens como: suporte para diversas linguagens de programação, ser uma ferramenta *open source*, tem uma grande variedade de métricas, é fácil de ser utilizada em projetos reais, é uma ferramenta popular tanto no meio acadêmico quanto no mercado, possui interface *web* que suporta o acompanhamento evolutivo das métricas e das diferentes versões dos projetos, gerar relatórios das métricas, é uma ferramenta extensível, existindo diversos *plug-ins* que podem ser usados para dar suporte às análises de código realizadas e incorpora três grandes ferramentas (PMD, CheckStyle, FindBugs) (SIEBRA, 2014).

É uma ferramenta que permite a customização de um conjunto de regras que podem ser utilizadas na análise de um projeto. Essas regras são divididas de acordo com sua severidade: muito altas, altas, médias, baixas e muito baixas. Possuem seis categorias (CAMPBELL, 2014):

- Bugs: são erros lógicos como falhas ao fechar arquivos, erros em conexão com bancos de dados, métodos que checam a igualdade mas sempre retornam falso (ou verdadeiro), entre outros;
- Bugs em potencial: são relacionados aos problemas condicionais que irão ocorrer constantemente como: checagem de valores nulos que levam a desreferenciação, operações matemáticas erradas, e entre outros;
- Indicadores de possíveis erros de programação: comparações que fazem atribuição ao invés de comparações retornando sempre verdadeiro, códigos nunca utilizados, e entre outros;



- Futuros erros de programação: tornam o sistema difícil de ser mantido como classes muito grandes que precisam ser refatoradas, métodos que são muito longos e/ou muito complexos, ausência de padrão de nomenclatura, e entre outros;
- Ineficiência: Impedem o programa de utilizar sua eficiência máxima, como: uso ineficientes de *strings*, *imports* desnecessários, código não utilizado, e entre outros;
- Inconsistência de estilo: regras relacionadas ao estilo de codificação, como posicionamento das chaves, indentação, e entre outros.

A *DashBoard* do *SonarQube* mostra os arquivos e pacotes com maiores problemas, o resultado das métricas analisadas no projeto, como por exemplo: cobertura de testes e duplicação; regras de codificação violadas, gravidade das regras violadas e entre outros aspectos (CAMPBELL, 2014).

Como toda ferramenta de análise estática, pode existir a possibilidade de falsos resultados, que indicam uma possível vulnerabilidade quando na verdade não existem. São os chamados falsos positivos. Podem existir também os falsos negativos, que são quando vulnerabilidades existentes não são identificadas pela ferramenta. O *SonarQube* possibilita a classificação dos erros encontrados como falsos positivos (LOURIDAS, 2006).

## 4. MATERIAIS E MÉTODOS

Este Capítulo apresenta primeiramente a descrição do plano de coleta de métricas que será realizado nos Capítulos 5 e 6. As outras seções do Capítulo caracterizam o objeto de estudo descrevendo as fases de execução do estudo de caso, apresentando os aspectos relacionados ao *Planejamento e desenho do estudo de caso* que envolve a caracterização do objeto de estudo, logo após a forma como os dados são coletados e por último como será realizada a *Análise dos dados* e a *Redação do Estudo*.

### 4.1. PLANO DE COLETA DE MÉTRICAS

Este trabalho realizou um plano de coleta de métricas com a utilização do GQM, que define objetivos, questões e métricas. A partir do GQM foram definidos três objetivos de medição, que norteiam este trabalho. Os objetivos de medição definidos foram: *avaliação da qualidade interna do código* (objetivo de medição 1), *avaliação da documentação disponível* (objetivo de medição 2) e *análise da quantidade de requisições de manutenção corretiva registradas* (objetivo de medição 3).

O método GQM utiliza um processo de medição em que os dados são: coletados, medidos, registrados, processados e interpretados. Na interpretação é verificado se os dados gerados respondem às questões definidas alcançando o objetivo de medição proposto pelo método.

O procedimento para a coleta de dados foi definido considerando alguns aspectos principais como: o responsável pela coleta das métricas, o período em que os dados foram coletados, e como os dados podem ser coletados eficiente e efetivamente. Para realizar esse procedimento, os dados foram coletados manualmente e também de forma automática através da utilização de ferramentas.

As métricas relacionadas à qualidade interna do código foram coletadas de forma automática. A definição dessas métricas surgiu devido a necessidade do próprio *Ministério X* de continuar trabalhos previamente realizados no órgão. Os trabalhos anteriores foram o de Crozara (2014) que definiu um perfil de qualidade através da configuração da ferramenta SonarQube para identificação de erros presentes no sistemas. O trabalho de Schaidt e Rodrigues (2015) deu continuidade

ao trabalho Cruzara (2014) e adicionou uma estratégia de coleta de métricas e métricas básicas.

Este trabalho utiliza as métricas já coletadas nos trabalhos de Cruzara (2014) e Schaidt e Rodrigues (2015) e incluiu novas métricas extraídas pela própria ferramenta, além de coletar métricas de novos sistemas. As métricas referentes à qualidade interna do código estão presentes no objetivo de medição 1, definido pelo método do GQM.

As métricas referentes à documentação dos sistemas foram definidas com base no trabalho de RAMOS (2004) que utiliza um conjunto de métricas de documentação de código para avaliar os sistemas legados de uma organização. Como o *Ministério X* não tinha clareza sobre as documentações dos seus sistemas, este trabalho veio solucionar esse problema por meios das métricas coletadas.

As métricas relacionadas à documentação estão presentes no objetivo de medição 2 e foram coletadas de forma manual, avaliando o conteúdo das documentações. As métricas definidas para esse objetivo visam obter diferentes aspectos da documentação mínima que um sistema deve possuir. Essa documentação mínima foi definida pelo próprio *Ministério X* e são: *modelo de banco de dados*, *descrição de casos de uso* e *documentação de regras de negócio*. Vale ressaltar que as métricas não abrangem outros tipos de documentação que podem, ou não, existir no repositório de documentações dos sistemas.

As métricas relacionadas às quantidades de manutenção corretiva surgiram de queixas dos integrantes do órgão sobre altos gastos com manutenção corretiva e falta de conhecimento sobre os defeitos registrados. Elas estão presentes no objetivo de medição 3 que avalia os registros de manutenção corretiva feitos pelos usuários dos sistemas a fim de obter clareza sobre a quantidade de requisições de manutenção corretiva demandadas pelos sistemas e entender a causa desses defeitos.

Para análise das requisições de manutenção corretiva dos sistemas foi feito um estudo sobre as descrições das requisições de manutenção corretiva registradas na ferramenta OTRS (*Open Source Ticket Request System*), após esse estudo foram coletado métricas manuais referentes aos diferentes aspectos presentes nessas requisições. Esse estudo realizado sobre as requisições de manutenção corretiva considerou os chamados abertos entre o dia 01 de abril de 2015 e 01 de setembro de 2015. Todos os chamados presentes na ferramenta OTRS nesse

período de tempo foram analisados e classificados de acordo com as métricas definidas para esse objetivo de medição.

Após a coleta das métricas referentes aos objetivos de medição definidos pelo GQM foi analisado a relação entre os valores obtidos através de gráficos. Para realizar esta análise foram utilizadas três variáveis principais: a *métrica RCI (Rules Compliance Metric)*, o índice de documentação dos sistemas e o número de manutenção corretiva.

O RCI é uma métrica automatizada extraída através da ferramenta SonarQube, e para seu cálculo é utilizado um conjunto de duas outras métricas extraídas pela própria ferramenta que são: os erros presentes no sistema e o número de linhas de código. Essas métricas fazem parte das métricas definidas e coletadas no objetivo de medição 1 do GQM.

O índice de documentação é uma métrica calculada de forma manual. Ela utiliza para o seu cálculo três métricas definidas e coletadas no objetivo de medição 2. Através das métricas já coletadas o índice de documentação gera um valor numérico de 0 a 1 que representa o quanto o sistema é documentado.

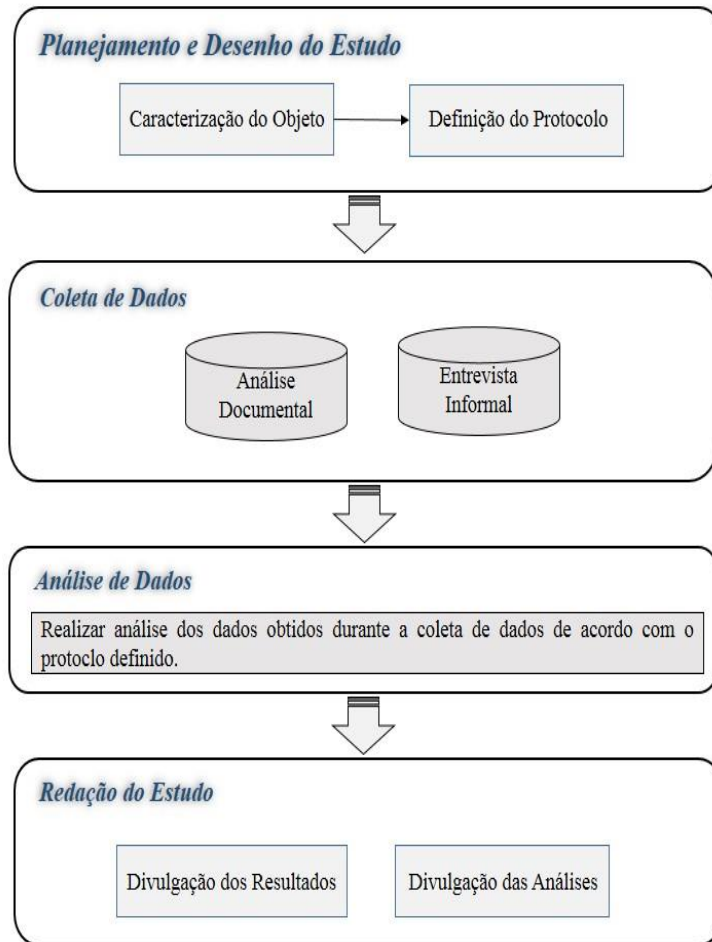
O número de manutenções corretivas utiliza a métrica coletada pelo objetivo de medição 3 que é o total de registro de manutenções corretivas presentes no OTRS durante o período de análise.

A matriz de decisão foi realizada utilizando o RCI em conjunto com os valores de criticidade dos sistemas para a organização. Essa matriz utiliza a combinação desses dois valores para cada sistema em um gráfico. O posicionamento dos sistemas neste gráfico determina qual é a melhor decisão a ser tomada para os sistemas legados.

## 4.2. ESTUDO DE CASO

Neste trabalho foi utilizado como referência o trabalho de Runeson (2009), que conduz o processo do desenvolvimento de um estudo de caso baseado em quatro grandes etapas de desenvolvimento, que são: *Planejar e Desenhar o Estudo de Caso*, que envolve o planejamento do estudo e a preparação, em que são realizadas a caracterização do objeto e a definição do protocolo; *Coletar Dados*, onde as técnicas de coleta são empregadas na execução do estudo para obtenção dos resultados; *Analisar Dados*, transformando-os em informações essenciais para a tomada de decisões; e *Relatar Estudo*, onde ocorre a redação de todo o processo do

estudo do caso (Runeson, 2008). Todas essas etapas foram utilizadas neste trabalho e estão ilustradas na Figura 6.



**Figura 6:** Etapas de execução do estudo de caso. Fonte: Autor.

#### 4.2.1 Planejamento e Desenho do Estudo de Caso

O planejamento do estudo de caso é a fase em que diversos elementos são investigados como: o objetivo que se pretende alcançar com o estudo de caso, o caso que está sendo estudado, o contexto em que o caso está inserido, o método utilizado para a coleta de dados e a estratégia para seleção e coleta de dados (Runeson, 2009).

##### 4.2.1.1. Caracterização do Objeto de Estudo

Este trabalho está inserido no contexto de uma organização da Administração Pública Federal, denominado de *Ministério X*. As áreas de competência desse

*Ministério* são os serviços de radiodifusão, postais e de telecomunicações. Todas essas áreas possuem sistemas que são mantidos pela TI.

A atividade de contratação para os serviços como o de radiodifusão tornou-se uma ferramenta de grande importância para o Ministério X, pois adiciona valor ao produto desenvolvido, melhorando os processos internos e os serviços prestados.

Para tais serviços de TI também é realizado a terceirização. Os três contratos mais significativos gerenciados pela área de TI são relacionados à fábrica de software, responsável pela manutenção e desenvolvimento de sistemas; à área de qualidade, com validação dos entregáveis pela fábrica e contagem de pontos de função; e infraestrutura de TI.

Atualmente o órgão possui um Termo de Cooperação com a UnB. Nesse termo encontra-se a definição de processos relacionados à manutenção e à catalogação de softwares legados, a aferição de qualidade desses softwares e abordagem para tomadas de decisões em relação aos mesmos. O órgão também passa por uma transição contratual.

No entanto, no antigo contrato de manutenção e desenvolvimento de serviços do Ministério X, consta que os serviços de manutenção prestados pela empresa contratada eram remunerados em razão do número de horas de serviço técnico consumidas (BRASIL, 2011).

Nesse mesmo contrato (BRASIL, 2011), para sistemas legados, eram previstas como serviço as atividades de evolução e de manutenção corretiva. Constavam-se que todos os sistemas construídos fora da arquitetura adotada pelo *Ministério X*, ou seja, os sistemas herdados de outros órgãos da APF, teriam que ser migrados para essa arquitetura; e a empresa contratada receberia por hora de serviços prestados. Nos demais casos, o serviço de manutenção corretiva não era remunerado.

No novo contrato (BRASIL, 2015), a atividade de manutenção corretiva é realizada por meio da emissão de ordens de serviços na modalidade de sustentação de sistemas e abertura de chamados pelos usuários do Ministério. Essa modalidade inclui os serviços de Apuração Especial, Suporte ao Usuário, Apoio à Produção, Documentação de Sistemas, Treinamento de Usuários e Manutenção corretiva. Já o faturamento dessa modalidade é realizado por meio de um valor fixo obtido a partir do tamanho funcional do sistema sustentado medido em pontos de função. Dessa

forma, o faturamento é realizado ao fim do período de sustentação, que vai do primeiro ao último dia do mês.

O novo contrato permite que a quantidade de sistemas sustentados possam sofrer alterações ao longo da execução do contrato, para menos ou para mais. Os sistemas que serão incluídos nesse contrato ainda serão definidos pelo *Ministério X* (BRASIL, 2015).

Após a análise dos problemas e necessidades presentes no *Ministério X*, o diagnóstico é que o órgão necessita de apoio para avaliar o atual estado de seus softwares legados. Isso foi realizado neste trabalho com o intuito de gerar uma caracterização do atual estado do legado, servindo como base para futuras decisões contratuais, onde se define os sistemas são candidatos a serem inclusos nesse novo contrato.

Para identificar a situação dos softwares legados avaliados no *Ministério X*, foi realizada a caracterização do órgão por meio das técnicas de coleta de dados documental e entrevistas informais. O documento utilizado foi o documento conjunto do plano estratégico TI (PETI) e plano diretor de TI (PDTI) (BRASIL, 2014b) para o período de 2013 a 2015 (BRASIL, 2014b). Na Tabela 1, encontra-se a caracterização do documento no contexto deste trabalho.

**Tabela 1:**Objetivos estratégicos específicos ao PETI. Fonte: (BRASIL, 2015, adaptado).

<b>Objetivos estratégicos específicos ao PETI</b>
OE-PETI 1. Promover a governança de TI no órgão.
OE-PETI 2. Evoluir no atendimento das áreas finalísticas do órgão, balanceando o portfólio de projetos e serviços a partir das diretrizes do planejamento estratégico do órgão.
OE-PETI 3. Aprimorar a gestão de TI no órgão.
OE-PETI 4. Redefinir a estrutura organizacional e a composição das equipes envolvidas nas atividades de TI do órgão.
OE-PETI 5. Melhorar continuamente os serviços de TI do órgão.
OE-PETI 6. Estabelecer e aprimorar a(s) arquitetura(s) de referência para a TI do órgão, de forma alinhada com as diretrizes do governo federal.
OE-PETI 7. Prover soluções confiáveis e disponíveis que oferecem recursos de: mobilidade, colaboração, desmaterialização de processos e transparência para com a sociedade.
OE-PETI 8. Promover a segurança da informação e comunicação do órgão.

Entre os objetivos estratégicos apresentados na Tabela 1, destaca-se o OE – PETI 5 que tem por função melhorar continuamente os serviços de TI no *Ministério X*, pois abrange a melhoria dos sistemas legados do órgão.

Para analisar o ambiente e os processos da organização, no documento foi realizada uma análise SWOT para identificar as forças, fraquezas, oportunidades e ameaças organizacionais. As oportunidades e ameaças têm origem no ambiente externo e a organização não exerce controle sobre as mesmas. Já as fraquezas e as forças são relacionadas com a realidade interna do *Ministério X*. As forças, fraquezas, oportunidades e ameaças que impactam neste trabalho estão em destaque na tabela na Tabela 2 que apresenta o resultado dessa análise.

**Tabela 2:** Análise SWOT da CGTI do Órgão. Fonte: (BRASIL, 2015, adaptado)

<b>Análise FFOA</b>	
<b>Forças</b>	<ul style="list-style-type: none"> <li>• <b>Equipe conhecedora do ambiente do órgão, das práticas boas e ruins implementadas no passado;</b></li> <li>• <b>Ambiente saudável e colaborativo;</b></li> <li>• Capacidade em contratos de terceirização flexíveis e aderentes à demanda de serviço;</li> <li>• Diagnóstico e plano de reestruturação da CGTI estabelecidos;</li> <li>• <b>Equipe comprometida;</b></li> <li>• <b>Equipe aberta a mudanças de processos e práticas.</b></li> </ul>
<b>Fraquezas</b>	<ul style="list-style-type: none"> <li>• Metodologias e processos de trabalho não definidos e/ou formalizados;</li> <li>• <b>Concentração da operação em serviços terceirizados, com risco de baixo desempenho ou substituição de fornecedores;</b></li> <li>• <b>Conhecimento e processos críticos concentrados na equipe dos fornecedores;</b></li> <li>• <b>Rotatividade das equipes dos fornecedores;</b></li> <li>• Elevado volume de contratos geridos e fiscalizados por número inadequado de servidores;</li> <li>• Instalações físicas inadequadas;</li> <li>• Baixa capacidade de planejar, especificar e gerir processos de aquisição;</li> <li>• <b>Arquitetura de TI heterogênea;</b></li> <li>• Portfólio de serviços desatualizado em relação às novas tecnologias de mobilidade, colaboração e automação de processos;</li> <li>• Quantitativo inadequado de servidores;</li> <li>• Baixa qualificação dos servidores em processos de governança e gestão de serviços de TI.</li> </ul>
<b>Oportunidades</b>	<ul style="list-style-type: none"> <li>• Capacidade de aquisição de produtos e serviços especializados;</li> <li>• <b>Orçamento adequado à necessidade de operação, sem muita pressão de contingenciamento;</b></li> <li>• <b>Apoio da SLTI na implantação de projetos e processos;</b></li> <li>• <b>Normativos, Acórdãos e solicitações de auditorias sustentam ações de melhoria de processos e equipes de TI;</b></li> <li>• <b>Patrocínio institucional para a implementação de mudanças nos serviços, estrutura, processos, pessoas da CGTI;</b></li> <li>• <b>Agilidade da alta gestão do Ministério na tomada de decisões;</b></li> <li>• Sensibilização por parte do órgão central das deficiências de pessoal e gestão da CGTI;</li> <li>• Servidores do Ministério e de outros órgãos interessados em trabalhar na CGTI do Ministério;</li> <li>• <b>TI é vista como crítica para implementação da estratégia do Ministério;</b></li> <li>• <b>Possibilidade de cooperação com universidades.</b></li> </ul>
<b>Ameaças</b>	<ul style="list-style-type: none"> <li>• Normativos, Acórdãos e solicitações de auditorias que direcionem contrariamente às ações de TI do Ministério;</li> <li>• <b>Percepção negativa dos usuários da qualidade dos serviços prestados e</b></li> </ul>



### Análise FFOA

- capacidade de entrega da CGTI;
- **Equipes de TI descentralizadas nas secretarias no Ministério;**
- **Backlog crescente de projetos e serviços de TI.**

#### 4.2.1.2. Fundamentação do trabalho

Para guiar a coleta de dados do estudo de caso selecionado, foi definida uma questão de pesquisa e três hipóteses. A questão de pesquisa deste trabalho foi desenvolvida a partir do problema organizacional identificado e descrito no tópico 1.2 do Capítulo 1 e é dada como se segue:

*Como avaliar sistemas legados a fim de apoiar uma organização em seus esforços de melhoria dos serviços de TI a partir de três aspectos: qualidade interna do código, documentação e registros de manutenção corretiva?*

A partir da questão de pesquisa foram definidas as seguintes hipóteses:

*H1. O volume de manutenção corretiva realizadas nos sistemas legados é inversamente proporcional a documentação;*

*H2. Os sistemas que possuem um alto número de requisições de manutenção corretiva possuem uma baixa qualidade interna do código;*

*H3. Os sistemas que possuem baixa qualidade interna do código possuem um baixo nível de documentação.*

#### 4.2.2 Coleta de dados

A técnica de coleta de dados pode ser dividida em três níveis. O primeiro nível se refere aos métodos diretos, esse é o nível em que o pesquisador está em contato direto com o tema e os dados coletados em tempo real, utilizando, como por exemplo, entrevistas e observações participativas. O segundo nível se refere aos métodos indiretos em que o pesquisador recolhe os dados sem interagir com os sujeitos durante a coleta utilizando ferramentas de engenharia de software. Já o terceiro nível se refere à análise de artefatos de trabalho já finalizados como: especificações de requisitos, relatórios de avaliação da organização ou até mesmo dados da contabilidade (Runeson, 2009).

Neste trabalho foram utilizados os três níveis de coleta de dados. Os métodos diretos foram utilizados com entrevistas informais. Os métodos indiretos se deram por meio das ferramentas SonarQube e o OTRS. Já a análise de artefatos ocorreu utilizando os documentos do plano estratégico TI (PETI) e plano diretor de TI (PDTI) além do documentação de modelo de dados, descrição de casos de uso e documento de regra de negócio dos softwares analisados.

#### 4.2.2.1 Análise Documental

A análise documental foi realizada utilizando documentos específicos do órgão, que são: o plano estratégico TI (PETI) e plano diretor de TI (PDTI) para o período de 2013 a 2015 (BRASIL, 2014b).

Além dos documentos do órgão foi realizado uma análise das informações sobre manutenção corretiva dos sistemas presentes na ferramenta OTRS. Os registros da ferramenta SonarQube foram utilizados para avaliar a qualidade interna dos sistemas. O repositório de projetos do Ministério X, foi uma fonte de análise documental para o acesso as documentação dos sistemas.

#### 4.2.2.2 Entrevista Informal

As entrevistas informais foram fontes de coleta de dados relacionados ao contexto do caso selecionado para estudo neste trabalho. Delas foram extraídas as questões referentes ao método GQM, assim como informações específicas do órgão.

Foram realizados oito encontros para a realização das entrevistas informais com duração de 1 horas e 30 min a 2 horas. O público-alvo para a realização das entrevistas foi: os analistas de sistemas, o gestor dos sistemas e o responsável pelo registro de chamados de serviços. As questões abordadas foram:

- Quais os principais aspectos da documentação precisam ser analisados;
- Quais os documentos essenciais para os sistemas legados;
- Qual a documentação mínima exigida pelo órgão;

- Quais desses documentos eram necessários serem avaliados;
- Quais os sistemas ativos dentro do Órgão;
- Onde e como eram realizados os registros de manutenção corretiva;
- Onde se localizava a documentação e o código fonte do sistema.

#### 4.2.2.3 Métricas

Neste trabalho foi realizado um levantamento sobre a atual situação dos *softwares* legados presentes no *Ministério X*. Esse levantamento foi efetivado utilizando o método GQM através de uma análise de três aspectos dos softwares legados: qualidade do código fonte, documentação e quantidade de defeitos registrados. A análise desses três aspectos teve como objetivo possibilitar uma maior visão do gestor sobre seus softwares legados.

Algumas questões foram realizadas no *Ministério X* para caracterizar e identificar a meta da organização. Como as questões apresentadas a seguir:

- **Objeto de Estudo:** o que será analisado?
- **Objetivo:** porque o objeto será analisado?
- **Enfoque de qualidade:** qual atributo do objeto será analisado?
- **No ponto de Vista:** quem usará os dados coletados?
- **Contexto:** em qual contexto será analisado?

Foi definida uma meta organizacional que iria servir para este trabalho, através da meta organizacional foram definidos os objetivos e questões do GQM. Das questões estabelecidas foram derivadas as métricas. As métricas são capazes de definir operacionalmente os dados quantitativos que serão coletadas para responder a essas questões.

Para promover a automatização de algumas métricas coletadas neste trabalho, foi utilizada a verificação, a inspeção do software e a análise estática de código através da ferramenta de análise estática, *SonarQube*. Essa ferramenta foi utilizada juntamente com a estratégia para a coleta de métricas relacionadas à qualidade de código definida no trabalho de Schaidt e Rodrigues(2015).

A realização da coleta das métricas foi feita através da execução de atividades. A Figura 7 relata as atividades que foram realizadas e seu fluxo de execução.



**Figura 7:** Fluxo de atividades realizadas. Fonte autor.

A primeira atividade foi a de definição de métricas. Essa atividade ocorreu a partir da necessidade de informação que o *Ministério X* precisava obter sobre seus sistemas legados. Essas necessidades de informação foram identificadas a partir das reuniões e do estudo da própria estrutura organizacional. Elas foram sanadas utilizando o método do GQM através da definição de objetivos, questões e métricas.

A fase de coleta de métrica é a aplicação do método GQM com a obtenção dos resultados das métricas coletadas sobre os softwares legados. Nessa fase são obtidos os dados quantitativos.

A avaliação dos resultados das métricas foi realizada entre o pesquisador e os membros do *Ministério X*. Na avaliação foi verificado se os dados coletados através da utilização do GQM atendem às necessidades de informação sobre o legado de software. Quando a conclusão dessa avaliação informou que as métricas coletadas não eram apropriadas, foram definidas novas métricas para uma nova coleta e avaliação dos dados. Quando concluiu-se que as métricas utilizadas eram apropriadas os seus resultados foram armazenados.

O armazenamento dos resultados foi realizado após a finalização da coleta das métricas e teve como propósito gerar uma base histórica de métricas e informações sobre os legados de softwares presentes no *Ministério X*.

#### 4.2.3 Análise dos dados

Durante a análise dos dados foi realizado um comparativo entre os dados coletados e a teoria apresentada neste trabalho. Essa comparação foi efetuada para verificar se os dados coletados estão coerentes com a teoria e verificar se os dados são convergentes uns com os outros.

Na fase de análise dos dados, o pesquisador e os membros do *Ministério X* verificaram se os dados coletados estavam atendendo as suas necessidades de informação sobre os softwares legados. Essa análise realizada tem como objetivo efetivar um estudo sobre os resultados das métricas coletadas, verificando se os dados são do tipo especificado e se o valor informado para uma determinada medida é possível.

Os dados obtidos por este trabalho são quantitativos, pois quantificam características dos sistemas legados relacionadas à qualidade interna do código, aspectos internos da documentação e registros de manutenção corretiva. Também são qualitativos, pois investigam o contexto específico em que o sistemas legados estão inseridos.

##### 4.2.3.1 Validação dos Dados

A validade do protocolo de construção é referente ao quanto o plano de estudo de caso foi construído com base em referências teóricas significativas ou teve os dados utilizados bem selecionados (EISENHARDT, 1989). Dessa forma, a falta de um levantamento bibliográfico estruturado nesta pesquisa pode ocasionar em uma fundamentação debilitada, enfraquecendo assim certas afirmações, ou seleções de métricas.

Validade interna do protocolo realizado é referente ao quanto os resultados ou conclusões do estudo correspondem a uma realidade reconhecida pelos próprios participantes do estudo de caso (YIN, 1985). Um problema que pode impactar a validade interna deste estudo é que não foram encontrados muitos trabalhos anteriores que apoiem ou fortaleçam as hipóteses apresentadas.

Validade externa se refere ao quanto os resultados obtidos numa pesquisa podem ser generalizados para o restante da população (CAMPBELL, 1979). Como os softwares legados presentes no órgão são diferentes dos softwares legados presentes em outros órgãos, os dados apresentados não podem ser difundidos para outras organizações, possuindo extensibilidade restrita ao próprio Ministério X.

#### 4.2.4 Redação do Estudo

A redação do estudo é a fase em que os resultados são comunicados. Ela é a principal fonte de garantia de qualidade do estudo (RUNESON, 2008).

Neste trabalho a redação do estudo foi conduzida da seguinte maneira: o *Capítulo 1 – Introdução* compreende com a contextualização, problemas e objetivos deste trabalho; os Capítulos 2 e 3, que abordam o referencial teórico a *Manutenção de Software e Medição e Métricas*; o *Capítulo 4 – Matérias e Métodos*; o *Capítulo 5 – Avaliação de sistemas legados*; o *Capítulo 6 – Comparação e análise dos dados*; e o *Capítulo 7 – Conclusões e Trabalhos Futuros* com a finalização deste trabalho. É obtido por meio deste documento um relatório sobre o Estudo de Caso realizado.

## 5 AVALIAÇÃO DOS SISTEMAS LEGADOS

A fim de diminuir futuros gastos com manutenções dos sistemas legados, o *Ministério X* precisa obter informações que possibilitem a caracterização dos seus sistemas. Este Capítulo apresenta esta caracterização por meio da utilização do método GQM que engloba de três aspectos principais dos sistemas que são: a qualidade interna do código, a documentação, e os registros de manutenção corretiva.

### 5.1 APLICAÇÃO DO GQM

O GQM é um modelo de coleta de métricas que define objetivos, questões e métricas. Os objetivos de medição devem ser identificados através do objetivo de negócio, que define o propósito da aplicação do GQM (BASILI, 1994).

A identificação do objetivo de negócio foi definido a partir da motivação deste trabalho que é: caracterizar os sistemas legados por meio de três aspectos principais (qualidade interna do código, a documentação e o número de defeitos registrados).

O Quadro 1 representa o objetivo de negócio. O objetivo de negócio é a base para definição dos objetivos e questões do GQM.

**Quadro 1:** Objetivo de negócio. Fonte: Autor.

**Possibilitar uma maior visão do gestor sobre diferentes aspectos dos seus sistemas legados para que a organização possa definir melhor os seus contratos de manutenção.**

A partir do objetivo de negócio estabelecido para este trabalho foram identificados objetivos que foram estruturados conforme o modelo proposto por Basili (1996). Esses objetivos foram definidos de acordo com o contexto em que os sistemas legados estão inseridos no *Ministério X*. Dessa maneira existe três objetivos principais. Sendo eles:

- **Objetivo 1:** avaliação da qualidade interna do código;
- **Objetivo 2:** avaliação da documentação disponível;
- **Objetivo 3:** análise da quantidade de requisições de manutenção corretiva registradas.

Desses objetivos foram derivadas questões. Para responder às questões, foram buscadas na literatura, métricas que fossem capazes fornecer as respostas.

O Quadro 2 representa o objetivo 1: avaliação da qualidade interna do código. Esse objetivo de medição visa analisar a qualidade interna do código, utilizando a ferramenta de análise estática *SonarQube* para a obtenção de métricas automatizadas.

**Quadro 2:** Objetivo 1: Avaliação da qualidade do código do sistema legado. Fonte autor.

<p><b>Analisar:</b> a qualidade interna dos sistemas;</p> <p><b>Com que propósito:</b> avaliar;</p> <p><b>Com respeito à:</b> qualidade interna do software;</p> <p><b>Sobre o ponto de vista:</b> do analista de sistemas e do desenvolvedor;</p> <p><b>No contexto do:</b> Ministério X.</p>
--

Na Tabela 3 são apresentadas as questões definidas para o objetivo de medição 1. Para esse objetivo de medição foram definidas quatro questões principais que ao serem respondidas através dos valores obtidos pelas métricas caracterizam a qualidade interna do código dos sistemas legados presentes no *Ministério X*.

**Tabela 3:** Questões definidas para atender o objetivo de medição. Fonte autor.

<b>Objetivo 1: Avaliação da qualidade do código do sistema legado.</b>	
1.	Qual o tamanho do sistema?
2.	Qual o nível de duplicação do sistema?
3.	Qual a complexidade do sistema?
4.	Qual a quantidade de erros encontrados no código?
5.	Qual a cobertura de testes do sistema?

A Tabela 4 é referente a questão número um do primeiro objetivo de medição. Ela possui uma métricas principal que visa obter o tamanho do sistema em linhas de código.



**Tabela 4:** Questão 1. Avaliação da qualidade. Fonte: (SONAR, 2015).

<b>Questão 1: Qual é o tamanho do sistema?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações</b>
<b>M 1.1)</b> Tamanho do sistema em LOC	$X = A$ Onde: A = Total de linhas de código do sistema	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.

A Tabela 5 é referente a questão número dois do primeiro objetivo de medição. Ela possui quatro métricas que visam obter a complexidade duplicação do sistema.

**Tabela 5:** Questão 2. Avaliação da qualidade. Fonte: (SONAR, 2015).

<b>Questão 2: Qual o nível de duplicação do sistema?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações</b>
<b>M 2.1)</b> Densidade de duplicação	$D = A/B * 100$ <b>Onde:</b> D= Densidade de duplicação A = Linhas duplicadas; B = Total de linhas.	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.
<b>M 2.2)</b> Arquivos duplicados	$X = A$ <b>Onde:</b> X= Total de arquivos duplicados; A = Números de arquivos envolvidos em uma duplicação.	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.
<b>M 2.3)</b> Linhas duplicadas	$X = A$ <b>Onde:</b> X= Total de linhas duplicadas; A = Números de arquivos envolvidos em uma duplicação.	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.
<b>M 2.4)</b> Blocos Duplicados	$X = A$ <b>Onde:</b> X= Total de blocos duplicados; A = Números de arquivos envolvidos em uma duplicação.	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.

A Tabela 6 é referente a questão três do primeiro objetivo de medição. Ela possui quatro métricas principais que visam obter a complexidade ciclomática do sistema.

**Tabela 6:** Questão 3. Avaliação da qualidade. Fonte: (SONAR, 2015).

<b>Questão 3: Qual a complexidade do sistema implementado?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observação</b>
<b>M 3.1)</b> Complexidade ciclomática total	$CCT = \sum(PR)/N$ <p><b>Onde:</b> CCT= Complexidade ciclomática total; PR = Todas as palavras reservadas da linguagem que indiquem definição de funções, estruturas condicionais e estruturas de repetições; N = Número total de arquivos existentes no projeto. Obs: Somatório realizado de 0 a n.</p>	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.
<b>M3.2)</b> Complexidade ciclomática por função	$CCF = \sum(PR)/NF$ <p><b>Onde:</b> CCF = Complexidade ciclomática por função; PR = Todas as palavras reservadas da linguagem que indiquem definição de funções, estruturas condicionais e estruturas de repetições; NF = Número total de funções existentes no projeto. Obs: Somatório realizado de 0 a n</p>	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.
<b>M 3.3)</b> Complexidade ciclomática por classe	$CCC = \sum(PR)/NC$ <p><b>Onde:</b> CCC= Complexidade ciclomática por classe; PR = Todas as palavras reservadas da linguagem que indiquem definição de funções, estruturas condicionais e estruturas de repetições; NC = Número total de classes existentes no projeto. Obs: Somatório realizado de 0 a n.</p>	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.
<b>M 3.4)</b> Complexidade ciclomática por arquivo	$CCA = \sum(PR)/NA$ <p><b>Onde:</b> CCA = Complexidade ciclomática por arquivo; PR = Todas as palavras reservadas da linguagem que indiquem definição de funções, estruturas condicionais e estruturas de repetições; NA = Número total de arquivos existentes no projeto. Obs: Somatório realizado de 0 a n.</p>	Métrica extraída através da ferramenta SonarQube. O cálculo é realizado pela própria ferramenta.

O trabalho de Crozara (2014) configurou um Perfil de Qualidade para a ferramenta SonarQube. O Perfil de Qualidade define o tipo de linguagem a ser analisada, a classificação das violações de regra identificadas, padrões de expressões regulares que devem estar presentes ou não no código, os projetos aos quais o Perfil de Qualidade é aplicado e etiquetas que podem ser associadas a cada uma das regras de violação de código definidas.

O Perfil de qualidade foi configurado de acordo com as necessidades do *Ministério X* e determina quais as violações de código são classificados com erros de gravidade alta, média, baixa e muito baixa. Essas violações surgiram através da customização da ferramenta SonarQube utilizando os *plugins FindBugs, Squid, PMD e Checkstyle*. Alguns desses plug-ins identificavam a mesma violação de código, para que o sonarQube não contabilizasse erros repetidos, Crozara (2014) retirou as violações que estavam duplicadas entres os *plugins* utilizados.

Crozara (2014) definiu características das violações para que elas fossem classificadas como erros de gravidade muita alta, alta, média, baixa e muito baixa. Erros com gravidade muito alta estão relacionados aos aspectos de segurança e vulnerabilidade do código. Erros de gravidade alta são más práticas de programação, corretude, bugs, bugs em potencial e indicativos de erros de programação. Erros com gravidade baixa e média são inconsistência de estilo, corretude, ineficiência e indicativos de erro de programação. Já os erros classificados com gravidade muito baixa são referentes à inconsistência de estilo.

A Tabela 7 é referente à questão quatro do primeiro objetivo de medição. Ela possui as métricas referentes aos erros com gravidade muito alta, alta, média, baixa e muito baixa.

**Tabela 7:** Questão 4. Avaliação da Qualidade. Fonte: (SONAR, 2015).

<b>Questão 4:</b> Qual a quantidade de erros encontrados no código?		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observação</b>
<b>M 4.1)</b> Números de erros detectado no código	$X=A$ Onde: A=Número total de erros presentes no código.	Métrica extraída com a utilização da ferramenta SonarQube.
<b>M 4.2)</b> Números de erros com gravidade muito alta	$X=A$ Onde: A=Número total de erros com gravidade muito alta presentes no código.	Métrica extraída com a utilização da ferramenta SonarQube.

Questão 4: Qual a quantidade de erros encontrados no código?		
Métrica	Fórmula	Observação
<b>M 4.3)</b> Números de erros com gravidade alta	X=A Onde: A=Número total de erros com gravidade alta presentes no código.	Métrica extraída com a utilização da ferramenta SonarQube.
<b>M 4.4)</b> Números de erros com gravidade média	X=A Onde: A=Número total de erros com gravidade média presentes no código.	Métrica extraída com a utilização da ferramenta SonarQube.
<b>M 4.5)</b> Números de erros com gravidade baixa	X=A Onde: A=Número total de erros com gravidade baixa presentes no código.	Métrica extraída com a utilização da ferramenta SonarQube.
<b>M 4.6)</b> Números de erros com gravidade muito baixa	X=A Onde: A=Número total de erros com gravidade muito baixa presentes no código.	Métrica extraída com a utilização da ferramenta SonarQube.

A Tabela 8 é referente à questão cinco do primeiro objetivo de medição. Ela possui uma métrica que visa obter a cobertura de testes do sistema.

**Tabela 8:** Questão 5. Avaliação da Qualidade. Fonte: (SONAR, 2015).

Questão 5: Qual a cobertura de testes do sistema?		
Métrica	Fórmula	Observação
<b>M 5.1)</b> Total de cobertura de testes do sistema	$Cobertura = \frac{(CT + CF + LC)}{(2 * B + EL)}$ <p>Onde:            CT = Ramos que foram avaliadas como 'true' pelo menos uma vez;            CF = Ramos que foram avaliadas como 'false' pelo menos uma vez;            LC = Linhas cobertas (linhas a cobrir – linhas descobertas);            B = Número total de ramos;            EL = Número total de linhas executáveis (linhas a cobrir).</p>	Métrica extraída com a utilização da ferramenta SonarQube.

O Quadro 3 representa o objetivo 2: avaliação da qualidade da documentação disponível. Esse objetivo de medição tem como propósito avaliar a documentação existentes nos sistemas legados presentes no *Ministério X*. Avaliando o quanto a documentação tem ajudado ou contribuído na manutenção do sistema, ou seja o quanto ela tem ajudado os mantenedores dos sistemas legados.

**Quadro 3:** Avaliação das documentações. Fonte: Autor.

<p><b>Analisar:</b> a documentação do sistema legado;</p> <p><b>Com que propósito:</b> avaliar;</p> <p><b>Com respeito ao:</b> conteúdo da documentação do software;</p> <p><b>Sobre o ponto de vista:</b> mantenedores do sistema;</p> <p><b>No contexto do:</b> Ministério X.</p>
---

Na Tabela 9 está presente a questão definida para atender ao objetivo de medição número 2. A questão definida visa avaliar o quanto as documentações dos sistemas podem ser entendidas pela equipe mantenedora.

**Tabela 9:** Questões definidas para o objetivo 2. Fonte: autor.

<b>Objetivo 2: Avaliação da qualidade da documentação disponível</b>	
1.	A documentação pode ser entendida pela equipe mantenedora?

As métricas encontradas para responder a essa questão são apresentadas na Tabela 10.

**Tabela 10:** Questão 1. Avaliação da documentação do sistema. Fonte: (RAMOS, 2004).

<b>Questão 1: A documentação pode ser entendida pela equipe mantenedora?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações para interpretação</b>
<b>M 1.1)</b> Porcentagem de elementos do modelo entidade relacionamento que estão documentados.	$X = ((A1 + B1) / (A + B)) * 100$ Onde: A = Total de entidades B = Total de atributos A1 = Total de entidades documentadas B1 = Total de atributos documentados	00% a 25% – Documentação insuficiente 25% a 50% - pouco documentado 50% a 80% moderadamente documentado; 80% a 100% - bem documentado.
<b>M 1.2)</b> Indicador do nível de detalhamento dos	$X = A$ Onde:	(3) Bem detalhado (2) Detalhado (1) Pouco detalhado

<b>Questão 1: A documentação pode ser entendida pela equipe mantenedora?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações para interpretação</b>
requisitos.	A= (3) Bem detalhado (2) Detalhado (1) Pouco detalhado (0) Insuficiente	(0) Insuficiente
<b>M 1.3)</b> Indicador de qualidade dos requisitos.	X = A  Onde: A = (3) Excelente qualidade (2) Boa qualidade (1) Pouca qualidade (0) Insuficiente	(3) Excelente qualidade (2) Boa qualidade (1) Pouca qualidade (0) Insuficiente
<b>M 1.4)</b> Porcentagem de descrição de requisitos funcionais que possuem descrição das regras de negócio	X = A/B * 100  Onde: A = Total de requisitos funcionais registrados na especificação de requisitos que possuem descrição das regras de negócio; B = Total de requisitos funcionais registrados na especificação de requisitos.	00% a 25% - insuficiente 25% a 50% - baixo grau de descrição. 50% a 80% - descrição moderada 80% a 100% - alto grau de descrição
<b>M 1.5)</b> Indicador de facilidade de entendimento da especificação de requisitos funcionais	X = A/B * 100  Onde: A = Total de requisitos documentados; B = Total de requisitos que podem ser compreendidos.	00% a 25% - insuficiente 25% a 50% - baixo grau de descrição 50% a 80% - descrição moderada 80% a 100% - alto grau de descrição

O Quadro 4 representa o objetivo de medição 3: analisar a quantidade de requisições de manutenção corretiva registradas. Esse objetivo de medição tem como propósito avaliar os registros de manutenção corretiva dos softwares legados. Visando obter dados sobre quantidade de defeitos apresentados pelos sistemas em um determinado período de tempo.

**Quadro 4:** Objetivo 3. Fonte: autor.

**Analisar:** as manutenções corretivas;

**Com que propósito:** analisar;

**Com respeito ao:** registro de manutenções corretivas;

**Sobre o ponto de vista:** gerencial e técnico;

**No contexto da:** organização de TI.

A finalidade principal desse objetivo de medição é avaliar diferentes aspectos dos defeitos registrados pelos chamados de manutenção corretiva feito pelos usuários dos sistemas. Ele avalia se o sistema realmente apresenta um grande número de erros, dentro do número de defeitos registrados, ou se esse número de defeitos é referentes a outros aspectos dos sistemas que não sejam falhas ou erros dos mesmo.

A Tabela 11 possui as questões referentes ao objetivo de medição número 3.

**Tabela 11:** Questões definidas para o objetivo 3. Fonte: autor.

<b>Objetivo 3: Análise das requisições de manutenção corretiva registradas.</b>	
1.	Qual o total de defeitos apresentados pelo sistema?
2.	Qual a classificação dos defeitos apresentados pelo sistema?

A Tabela 12 representa a primeira questão do objetivo de medição 3. Ela possui como propósito obter o total de defeitos apresentados pelo sistema.

**Tabela 12:** Questão 1. Classificação de defeitos. Fonte: Autor.

<b>Questão 1: Qual o total de defeitos apresentados pelo sistema?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações para interpretação</b>
<b>M 1.1)</b> Total de erros registrados	$X = A$  Onde: A = Número de manutenções corretivas em determinado período.	Quanto maior o valor de X maior o número de defeitos apresentados pelo sistema.

A Tabela 13 representa a segunda questão do objetivo de medição 3. Ela possui como propósito identificar quais os principais defeitos apresentados pelo sistema.

**Tabela 13:** Questão 2. Classificação de defeitos. Fonte: Autor.

<b>Questão 2: Qual a classificação dos defeitos apresentados pelo sistema?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações</b>
<b>M 2.1)</b> Total de erros apresentados pelas requisições de manutenção corretiva.	$X = A$  <b>Onde:</b> A = Número de requisições de manutenção corretiva que são erros do sistema.	Quanto maior o valor de X maior o número de defeitos apresentados pelo sistema.

<b>Questão 2: Qual a classificação dos defeitos apresentados pelo sistema?</b>		
<b>Métrica</b>	<b>Fórmula</b>	<b>Observações</b>
<b>M 2.2)</b> Total de chamados de manutenção corretiva que estão duplicados.	$X = A$ <b>Onde:</b> A = É o número de requisições de manutenção corretiva duplicados, ou seja o problema já foi apresentado e solucionado em outra requisição.	Chamados duplicados apresentam defeitos que já foram resolvidos em um chamado anterior.
<b>M 2.3)</b> Total de chamados de manutenção corretiva que estão classificados errados.	$X = A$ <b>Onde:</b> A = É o número de requisições de manutenção corretiva que eram outro tipo de classificação mas foram registrados como manutenção corretiva de forma equivocada.	Chamados classificados errados representam chamados que não deveriam estar registrados como manutenção corretiva.
<b>M 2.4)</b> Total de chamados de manutenção corretiva que não foram realizados pois a descrição do problema era inconclusiva.	$X = A$ <b>Onde:</b> A = É o número de requisições de manutenção corretiva que não foram solucionadas pois a sua descrição era insuficiente.	Chamados abertos onde o usuário não descreve o erro de forma consistente, sendo necessário esclarecer as informações apresentadas para que a equipe mantenedora entenda a demanda registrada.
<b>M 2.5)</b> Total de chamados de manutenção corretiva que o erro apresentado na demanda não existe	$X = A$ <b>Onde:</b> A = É o número de requisições de manutenção corretiva em que o erro apresentado pelo requisitante não existe.	Chamados abertos em que o erro apresentado pelo usuário não existe.

### 5.1.1 Coleta e Análise dos dados

As métricas referentes aos três objetivos de medição definidos neste trabalho foram aplicadas aos 34 sistemas ativos no *Ministério X*. A lista de sistemas ativos ao qual as medições foram realizadas constam na Tabela 14.

**Tabela 14:** Lista de sistemas ativos. Fonte: autor.

<b>Nomes dos sistemas ativos no órgão</b>	
1	Cadastro de Usuários Externos (CADSEI)
2	Comissão Especial de Licitação (CEL)
3	Controle de processo e Documentos (CPROD)
4	Controle de visitantes (SCV)
5	Dados do setor de comunicação (DSCOM)



	<b>Nomes dos sistemas ativos no órgão</b>
6	Denúncia Web
7	Fale com o ministério (ouvidoriaWeb)
8	Geoserve (GEO)
9	Gestão de Processos e documentos (GPD)
10	Open Source Ticket Request System (OTRS)
11	Ponto Eletrônico
12	Radiofusão técnica (RADTEC)
13	SARF
14	Seg Ação
15	SEG Web
16	SELIN
17	Sistema corporativo (Siscorporativo)
18	Sistema de acessória parlamenta (ASPAR)
19	Sistema de acompanhamento de radiofusão (SICAF)
20	Sistema de arrecadação do fundel (SAF)
21	Sistema de compromisso cerimonial e agenda (SCCA)
22	Sistema de Consulta (SICAP)
23	Sistema de consulta de processo (SCP)
24	Sistema de gerenciamento de documentos (SGD)
25	Sistema de governo eletrônico registro e atendimento ao cidadão (SIGGESAC)
26	Sistema de pastas Funcionais, Jurídicas e Técnicas (SISPA)
27	Sistema de Permissão (SIP)
28	Sistema de radiofusão comunitária (RADCOM)
29	Sistema de recursos humanos (SRH)
30	Sistema de regime especial de tributação do plano nacional de banda larga (REPUBL)
31	Sistema de registro e análise de proposições de documentos (SRAPD)
32	Sistema Eletrônico de informação (SEI)
33	Sítio Conexão Minicom (Conexão minicom)
34	Sítio Intranet

#### 5.1.1.1 Objetivo de medição 1: Avaliação da qualidade interna do código do sistema legado

Nesta seção será apresentada a forma como foi realizada a coleta de dados com o objetivo de avaliar a qualidade de código dos sistemas legados. A avaliação do aspecto de qualidade interna dos sistemas serve como apoio para uma futura inclusão do sistema no contrato de manutenção. Pois de acordo com Bakota et al. (2014) sistemas com uma baixa qualidade interna do código, em alguma fase do seu ciclo de vida irá apresentar uma alta demanda por manutenção. Mesmo que atualmente o sistema não apresente um número significativo de requisições de manutenção corretiva que são realmente erros do sistema, eles podem ou não ser incluídos no contrato a partir desse indicador de qualidade interna.

A avaliação da qualidade interna utilizou a ferramenta SonarQube, por meio dela foram obtidos os valores das métricas que serão apresentados. A ferramenta não possui suporte para análise estática do código em projetos de linguagem ASP. Devido a essa restrição, os sistemas SARF, GPD e CEL construídos nessa linguagem não possuem as métricas selecionadas para esse objetivo de medição.

Já para projetos construídos em PERL era necessário comprar um plug-in para a ferramenta realizar a análise estática do código. Devido ao alto custo do plug-in e somente existir um projeto no Ministério X construído nessa linguagem, não foi viável realizar essa compra. Dessa maneira, não foi possível obter valores para as métricas relacionadas à qualidade do código para o sistema OTRS.

Também não foi possível obter as métricas desse objetivo de medição para o sistema sitio intranet, pois não existia o código fonte do sistema dentro do repositório de projetos do *Ministério X*. Onde deveria estar localizado o código desse projeto todas as pastas estavam vazias.

Dessa maneira, os seguintes sistemas não possuem métricas relacionadas ao objetivo de medição número 1: SARF, GPD e CEL, OTRS e sitio intranet.

### Questão 1: Qual o tamanho do sistema?

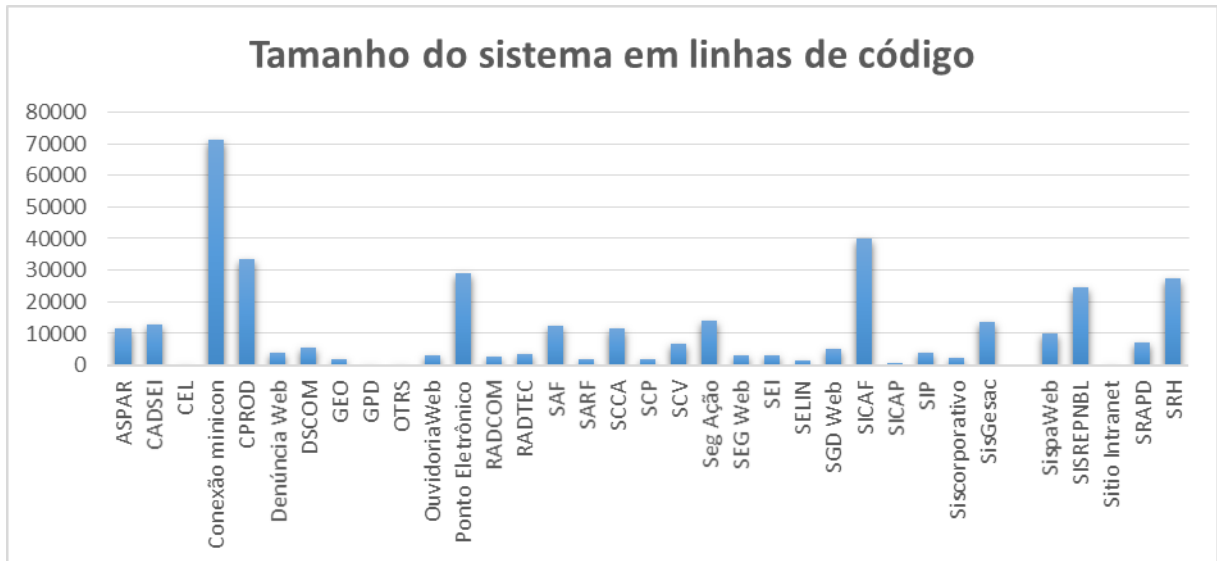
A Tabela 15 representa a Questão 1 desse objetivo de medição. A métrica referente a essa questão representa o tamanho dos sistemas em linhas de código. Esta Tabela contém a relação dos sistemas com seu tamanho de linha de código.

**Tabela 15:** Primeira métrica, tamanho do sistema em linhas de código. Fonte: Autor.

Questão 1	M 1.1) Tamanho do sistema em LOC
ASPAR	11.702
CADSEI	12.751
Conexão minicon	71.052
CPROD	33.643
Denúncia Web	4.033
DSCOM	5.618
GEO	1.965

<b>Questão 1</b>	<b>M 1.1) Tamanho do sistema em LOC</b>
OuvidoriaWeb	3.158
Ponto Eletrônico	28.887
RADCOM	2.831
RADTEC	3.372
REPNBL	24.461
SAF	12.359
SARF	1.742
SCCA	11.702
SCP	1.742
SCV	6.844
Seg Ação	13.943
SEG Web	3.226
SEI	3.063
SELIN	1.306
SGD	5.086
SICAF	40.002
SICAP	617
SIP	3.899
Siscorporativo	2.167
SISGESAC	13.527
SISPA	10.069
SRAPD	7.215
SRH	27.368

A Figura 8 utiliza os valores da Tabela 15 para expressar o tamanho do sistema em linhas do código. Pode-se observar que de acordo com os valores extraídos com a utilização ferramenta SonarQube os maiores sistemas em quantidade de linhas de código são: o conexão minicon, SICAF, ponto eletrônico, CPROD, SRH e REPNBL.



**Figura 8:** Tamanho de sistemas em linhas de código. Fonte: Autor.

## Questão 2: Qual o nível de duplicação do sistema?

A Questão 2 do objetivo de medição de avaliação da qualidade do código traz as métricas relacionadas a duplicação de código. Na Figura 22 são encontrados os valores obtidos através das métricas de duplicação de código para cada sistema.

**Tabela 16:** Segunda métrica do primeiro objetivo de medição. Fonte: Autor.

Questão 2	M 2.1) Densidade de duplicação	M 2.2) Arquivos duplicados	M 2.3) Linhas duplicadas	M 2.4) Blocos Duplicados
ASPAR	11,8%	36	2516	107
CADSEI	13,2%	35	2182	111
Conexão minicom	8,2%	138	10738	512
CPROD	26,5%	89	11699	482
Denúncia Web	0,9%	2	46	4
DSCOM	23,2%	34	2107	97
GEO	10,5%	8	308	12
ouvidoriaWeb	0,0%	0	0	0
Ponto Eletrônico	9,7%	62	4170	160
RADCOM	0,5%	2	22	2

<b>Questão 2</b>	<b>M 2.1) Densidade de duplicação</b>	<b>M 2.2) Arquivos duplicados</b>	<b>M 2.3) Linhas duplicadas</b>	<b>M 2.4) Blocos Duplicados</b>
RADTEC	6,7%	3	290	18
REPUBL	7,4%	46	2409	178
SAF	17,8%	40	2929	106
SARF	2,8%	3	80	4
SCCA	11,8%	36	2516	107
SCP	2,8%	3	80	4
SCV	27,6%	35	3306	199
Seg Ação	19,1%	68	3599	537
SEG Web	22,6%	19	1003	102
SEI	44,0%	14	1949	379
SELIN	74,0%	29	3251	214
SGD	12,9%	17	1000	53
SICAF	44,2%	157	28341	2752
SICAP	53,4%	6	721	18
SIP	41,0%	15	2403	361
Siscorporativo	18,0%	12	512	18
SISGESAC	26,7%	43	4413	249
SISPA	11,4%	42	1605	63
SRAPD	11,7%	16	1078	64
SRH	10,2%	60	4144	160

Neste trabalho os sistemas foram classificados quanto a sua densidade de duplicação de código, essa classificação obteve quatro valores para densidade de código sendo eles: baixa, média, alta e muito alta.

Foi definido que sistemas com densidade de duplicação de código baixa possuem valores entre 0% a 10%, densidade de duplicação média são sistemas que possuem mais que 10% e menos que 20%, sistemas com densidade de duplicação alta possuem valores de densidade de duplicação maior que 20% até 30%, já sistemas com densidade de duplicação muito alta possuem valores maiores do que 30%. A Tabela 17 possui a descrição desses valores.

**Tabela 17:** Critérios de classificação de duplicação do sistema. Fonte: Autor.

Densidade de Duplicação	Status	Valor de referência
Baixa	Ótimo	$0\% \leq DD \leq 10\%$
Média	Bom	$10\% < DD < 20\%$
Alta	Ruim	$20\% < DD < 30\%$
Muito Alta	Péssimo	$30\% \leq DD$

Na Tabela 18 encontra-se os sistemas presentes em cada um dos status definido na Tabela 17, de acordo com os valores de referência também explicitados nessa na mesma tabela. Utilizando como base a métrica 2.1 (densidade de duplicação) presente na Tabela 16.

**Tabela 18:** Sistemas pertencentes a cada classificação. Fonte: Autor.

Ótimos	Bons	Ruins	Péssimos
Conexão Minicom Denúncia Web OuvidoriaWeb Ponto eletrônico RADCOM RADTEC REPNBL SARF SCP	ASPAR CADSEI DSCOM GEO SAF SCCA Seg ação SGD SipaWeb SisCorporativo SRAPD SRH	CPROD DSCOM SCV Seg web SISGESAC	SEI SELIN SICAF SICAP SIP

### Questão 3: Qual a complexidade do sistema implementado?

Na ferramenta SonarQube a complexidade ciclomática é calculada utilizando as palavras chaves presentes no código como if, else, while, for, return. Para cada palavra chave é adicionado à complexidade da função o valor de +1, assim como laços de repetição que são contabilizados pela quantidade de repetições a serem executadas. Qualquer método ou função sempre terá no mínimo complexidade igual à 1 por causa de sua declaração public void, public int entre outras.

Os valores obtidos através das métricas de complexidade ciclomática definidas para a questão 3 desse objetivo de medição estão presentes na Tabela 19.

**Tabela 19:** Terceira questão do objetivo de medição 1. Fonte: Autor.

<b>Questão 3</b>	<b>M 3.1) Complexidade ciclomática</b>	<b>M3.2) Complexidade ciclomática por Método</b>	<b>M 3.3) Complexidade ciclomática por classe</b>	<b>M 3.4) Complexidade ciclomática por arquivo</b>
ASPAR	1802	2,1	8,2	8,3
CADSEI	2614	3,4	28,7	29,0
Conexão minicom	12294	2,2	10,1	10,0
CPROD	5949	2,4	27,0	26,9
DenúnciaWeb	542	2,6	10,4	10,4
DSCOM	1041	3,9	23,6	15,1
GEO	246	1,1	8,5	8,5
OuvidoriaWeb	586	2,2	12,5	12,5
Ponto Eletrônico	4095	2,0	9,8	9,8
RADCOM	279	2,2	4,2	4,2
RADTEC	685	3,8	21,4	21,4
REPUBL	4758	3,3	27,2	27,2
SAF	1869	2,7	15,8	15,8
SCCA	1802	2,1	8,2	8,3
SCP	297	1,7	10,6	14,1
SCV	1154	4,3	34,9	15,2
Seg Ação	2839	2,5	15,3	15,3
SEG Web	673	2,4	14,6	14,6
SEI	728	4,7	40,4	40,4
SELIN	203	25,4	3,7	0
SGD	796	1,9	10,0	10,0
SICAF	5318	2,0	17,4	18,5
SICAP	70	6,0	6,0	8,8
SIP	973	5,0	36,0	36,0
Siscorporativo	342	1,8	9,0	9,0
SISGESAC	3424	4,2	41,3	42,3
SISPA	1388	1,4	11,5	11,6
SRAPD	1252	2,3	16,7	16,7
SRH	3884	2,1	9,8	9,8

Utilizando os valores de referência presentes no trabalho de Schaidt e Rodrigues (2015) para a complexidade ciclomática por método é considera os seguintes dados: sistemas considerados ótimos possuem complexidade por método inferior a 5, sistemas bons entre 5 e menor que 10, sistemas ruins maior 10 e menor que 20, já sistemas considerados péssimos a complexidade ciclomática por método é maior que 20.

Foi definido neste trabalho que sistemas com status ótimo são sistemas com complexidade ciclomática por método muito baixa, bons com complexidade baixa, ruins com complexidade média e péssimos com complexidade alta.

A Tabela 20 possui os valores referentes para cada status definidos.

**Tabela 20:** Valores de complexidade dos sistemas. Fonte: (adaptado, Schaidt, Rodrigues; 2015).

Complexidade por método	Status	Valor de referência
Muito baixa	Ótimo	$0 \leq CC < 5$
Baixa	Bom	$5 \leq CC < 10$
Média	Ruim	$10 \leq CC < 20$
Alta	Péssimo	$20 \leq CC$

A Tabela 21 possui os sistemas de acordo com os status definidos para complexidade ciclomática por método. Utilizando como base a métrica 3.2: complexidade ciclomática por método. Essa métrica está presente na Tabela 19.

**Tabela 21:** Sistemas pertencentes as classificações de complexidade. Fonte: Autor.

Ótimos	Bons	Ruins	Péssimos
ASPAR CADSEI Conexão Minicom CPROD Denúncia Web DSCOM GEO OuvidoriaWeb Ponto Eletrônico RADCOM RADTEC REPNBL SAF SCCA SCP	SIP	-	SELIN



Ótimos	Bons	Ruins	Péssimos
SCV Seg AÇÃO Seg web SEI SGD SICAF SICAP Siscoporativo SIGESAC SISPA SRAPD SRH			

#### Questão 4: Qual a quantidade de erros encontrados no código?

O trabalho de Crozara (2014) realizou uma customização de regras na ferramenta SonarQube conforme a necessidade do *Ministério X*. Essa customização define o que seriam erros classificados com gravidade muito alta, alta, média, baixa e muito baixa.

De acordo com Crozara (2014) erros com gravidade muito alta estão relacionados aos aspectos de segurança e vulnerabilidade do código, gravidade alta são más práticas de programação, corretude, bugs, bugs em potencial e indicativos de erros de programação, erros com gravidade baixa são inconsistência de estilo, corretude, ineficiência e indicativos de erro de programação, já erros classificados com gravidade muito baixa são referentes a inconsistência de estilo.

Na Tabela 22 são encontrados os valores dos erros de gravidade muito alta, alta, média, baixa e muito baixa de cada sistema.

**Tabela 22:** Quarta questão do objetivo de medição 1. Fonte: Autor.

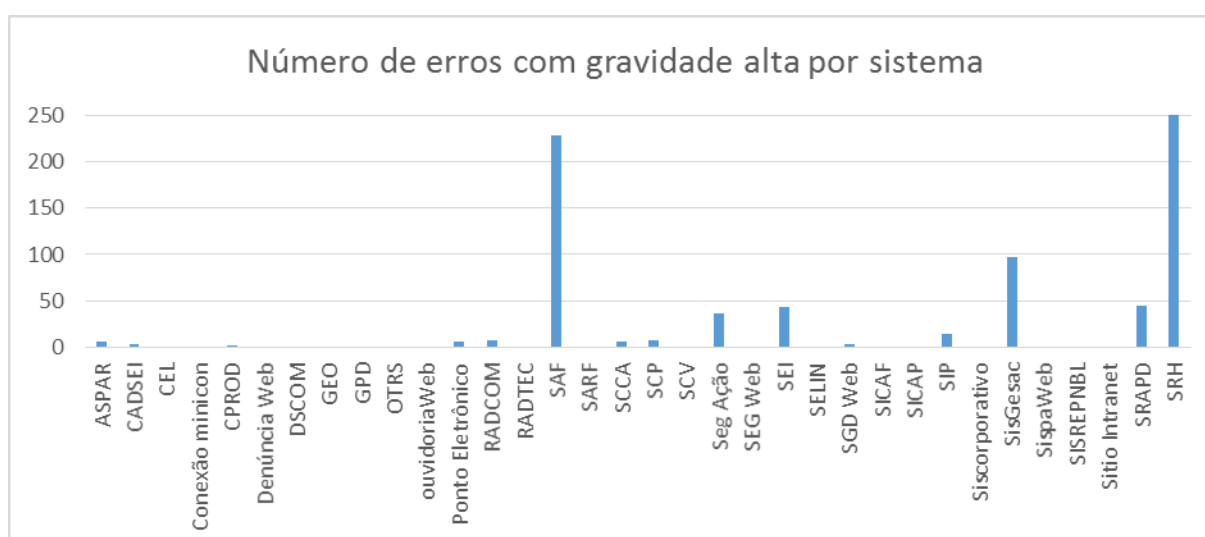
Questão 4	M 4.1) Números de erros detectado no código	M 4.1) Números de erros com gravidade muito alta.	M 4.2) Números de erros com gravidade alta	M 4.3) Números de erros com gravidade média	M 4.4) Números de erros com gravidade baixa	M 4.4) Números de erros com gravidade muito baixa
ASPAR	10404	0	6	479	4214	911
CADSEI	14784	0	4	992	5388	738
Conexão minicom	48508	0	0	15072	3292	603

<b>Questão 4</b>	<b>M 4.1) Números de erros detectado no código</b>	<b>M 4.1) Números de erros com gravidade muito alta.</b>	<b>M 4.2) Números de erros com gravidade alta</b>	<b>M 4.3) Números de erros com gravidade média</b>	<b>M 4.4) Números de erros com gravidade baixa</b>	<b>M 4.4) Números de erros com gravidade muito baixa</b>
CPROD	40332	0	1	2820	14521	1166
Denúncia Web	3676	0	0	97	1644	172
DSCOM	5642	0	0	458	1905	11
GEO	14882	0	0	66	609	77
GPD	74	0	0	3	31	0
Ouvidoria Web	65	0	0	17	14	0
Ponto Eletrônico	27336	0	6	1251	11136	1852
RADCOM	4674	0	8	667	2633	181
RADTEC	252	0	0	3	120	608
REPUBL	24984	0	0	1457	9578	1493
SAF	29295	0	228	5221	12492	567
SCCA	10404	0	6	479	4214	911
SCP	3911	0	7	649	1929	79
SCV	6814	0	0	552	2303	87
Seg Ação	12688	0	37	729	4701	692
SEG Web	2728	0	0	161	1042	169
SEI	6994	0	44	1303	2865	183
SELIN	3383	0	0	829	896	7
SGD	4386	0	4	231	1711	259
SICAF	40702	0	0	4315	11721	3333
SICAP	540	0	0	150	90	4
SIP	4514	0	15	104	1974	311
Siscorpora tivo	1578	0	0	62	665	72
SISGESA C	26340	0	97	4326	12877	699
SISPA	10606	0	0	970	3363	479
SRAPD	13100	0	45	1923	7106	308

Questão 4	M 4.1) Números de erros detectado no código	M 4.1) Números de erros com gravidade muito alta.	M 4.2) Números de erros com gravidade alta	M 4.3) Números de erros com gravidade média	M 4.4) Números de erros com gravidade baixa	M 4.4) Números de erros com gravidade muito baixa
SRH	47554	0	255	7033	25180	1195

Analisando os dados presentes na Tabela 22, pode-se observar que nenhum sistema possui erro classificado com gravidade muito alta. Já para os erros com gravidade alta existem sistemas que se destacam possuindo mais de 200 erros. Esses sistemas são o SAF com 228 erros com gravidade alta e o SRH com 255.

A Figura 9 possui o número de violações com gravidade alta presentes nos sistemas.



**Figura 9:** Sistemas com gravidade alta. Fonte: Autor.

### Questão 5: Qual a cobertura de testes do sistema?

A questão 5 desse objetivo de medição avalia o percentual de testes unitários existentes nos sistemas legados do *Ministério X*.

A ferramenta SonarQube só estava configurada para obter o percentual de testes unitários para sistemas em grails, nas outras linguagens a ferramenta somente identificava se o sistema possuía alguma classe de teste. Como em todos os outros sistemas a ferramenta identificou que não havia classes de testes foi

verificado manualmente no código fonte de cada projeto a existência de testes unitários. Conclui que não havia nenhum sistema com testes unitários com exceção do ouvidoriaWeb, por isso todos os valores de testes unitários dos outros sistemas presentes na Tabela 23 foram considerados como 0%.

Os valores referentes aos testes unitários dos sistemas estão presentes na Tabela 23.

**Tabela 23:** Quinta questão do objetivo de medição 1. Fonte: Autor.

<b>Questão 5</b>	<b>M 1.1) Total de cobertura de testes do sistema</b>
ASPAR	0%
CADSEI	0%
Conexão minicom	0%
CPROD	0%
Denúncia Web	0%
DSCOM	0%
GEO	0%
ouvidoriaWeb	81,6%
Ponto Eletrônico	0%
RADCOM	0%
RADTEC	0%
REPUBL	0%
SAF	0%
SCCA	0%
SCP	0%
SCV	0%
SEG Web	0%
SEI	0%
SELIN	0%
SGD	0%
SICAF	0%
SICAP	0%
SIP	0%
Siscorporativo	0%
SISGESAC	0%

<b>Questão 5</b>	<b>M 1.1) Total de cobertura de testes do sistema</b>
SISPA	0%
SRAPD	0%
SRH	0%

Utilizando os valores de referência presentes no trabalho de Schaidt e Rodrigues (2015) foi definido que sistemas com o valor de cobertura de testes maior ou igual a 80% são considerados ótimos, igual a 60% menor que 80% são sistemas bons, maior ou igual a 30% e menor que 60% ruins e menor que 30% são péssimos. Esses valores estão expostos na Tabela 24.

**Tabela 24:** Valores de referência para cobertura de testes. Fonte: (Adaptado, Schaidt, Rodrigues; 2015).

<b>Status</b>	<b>Valor de referência</b>
Ótimo	$CT \leq 80\%$
Bom	$60\% \leq CT < 80\%$
Ruim	$30\% \leq CT < 60\%$
Péssimo	$30\% < CT$

Como o sistema OuvidoriaWeb possui 81,6% de testes unitários é considerado um sistema ótimo, porém como todos outros sistemas possuem 0% de testes unitários são considerados sistemas péssimos. Como representa a Tabela 25.

**Tabela 25:** Relação de sistemas com suas classificações. Fonte: Autor.

<b>Ótimos</b>	<b>Bons</b>	<b>Ruins</b>	<b>Péssimos</b>
OuvidoriaWeb	-	-	ASPAR CADSEI Conexão Minicom CPROD Denúncia Web DSCOM GEO OuvidoriaWeb Ponto Eletrônico RADCOM RADTEC REPNBL SAF SCCA SCP SCV Seg AÇÃO

Ótimos	Bons	Ruins	Péssimos
			Seg web SEI SELIN SGD SICAF SICAP SIP Siscoporativo SISGESAC SISPA SRAPD SRH

#### 5.1.1.2 Objetivo de medição 2: Avaliação da documentação disponível

Nesta seção é apresentado a forma como foi realizada a coleta de dados com o objetivo de avaliar a documentação disponível dos sistemas legados. A avaliação da documentação dos sistemas legados visa obter um retorno de diferentes aspectos da documentação dos sistemas. Realizando uma análise de o quanto a documentação facilita a atividade de manutenção dos sistemas.

Essa análise foi feita sobre o conteúdo da documentação nos aspectos de completude, além de rastrear se os sistemas possuem ou não documentação. Essa visão mais aprofundada traz indicadores que ao serem utilizados pelo Ministério X justificam demandas para criação ou aperfeiçoamento de documentos. Essas novas demandas terão como resultados documentos que facilitam a compreensão dos sistemas do ponto de vista do mantenedor. Essa maior compreensão do sistema evita que o mantenedor insira novos erros ao realizar uma manutenção. Já que segundo Bakota et al. (2014) a atividade de manutenção de software pode causar a inserção de novos erros no sistema.

A avaliação da documentação foi realizada sem o conhecimento sobre os sistemas e sem contato com os usuários, com a equipe de manutenção, ou seja os resultados obtidos são provenientes apenas da leitura dos documentos disponibilizados para a avaliação. O registro desses resultados foi realizado manualmente através da coleta de métricas.

No *Ministério X* existe um conjunto de documentações que são consideradas as documentações mínimas que um sistema deve possuir. Sendo elas: o modelo de

banco de dados, descrição de casos de uso e a documentação de regras de negócio do sistema. As questões e métricas definidas nesse objetivo de medição possuem como propósito avaliar a completude dessas documentações. Tendo como foco principal o conteúdo presente nas mesmas.

Analisando as documentações mínimas de cada sistema foi observado que não são todos os sistemas que possui esse conjunto de documentação. Dessa maneira foi realizado um levantamento das documentações existentes de cada sistema. A Tabela 26 mostra o nome dos sistemas e as documentações que esse sistema possui.

**Tabela 26:** Relação dos sistemas com suas documentações existentes. Fonte: Autor.

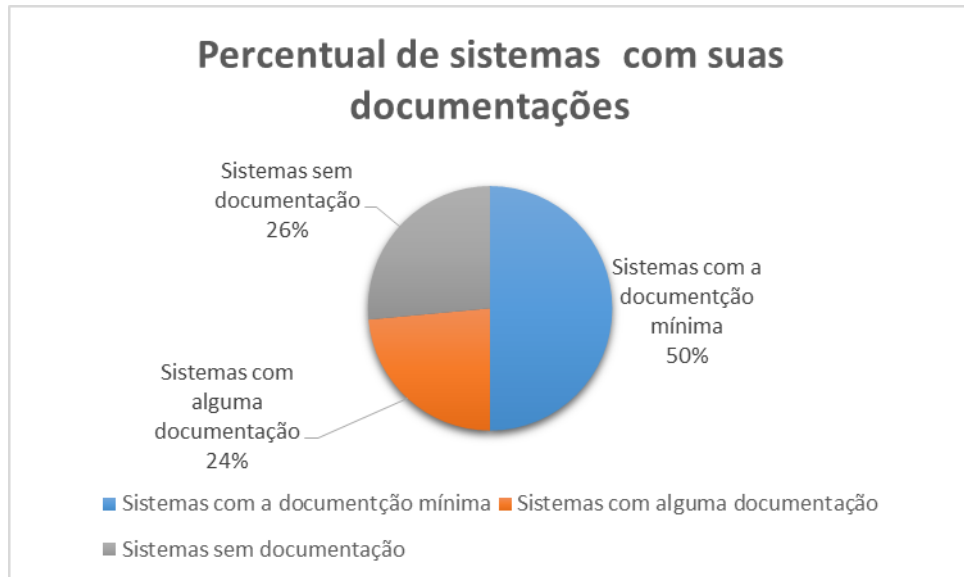
<b>Nome do sistema</b>	<b>Documentação existente</b>
ASPAR	Descrição de casos de uso Modelo de banco de dados Regras de negócio
CADSEI	Descrição de casos de uso Regras de negócio
CEL	
CPROD	Descrição de casos de uso Modelo de banco de dados Regras de negócio
Denúncia Web	Descrição de casos de uso Modelo de banco de dados Regras de negócio
DSCOM	Descrição de casos de uso Modelo de banco de dados Regras de negócio
GEO	
GPD	
OTRS	Modelo de banco de dados
OuvidoriaWeb	Descrição de casos de uso Modelo de banco de dados Regras de negócio
Ponto Eletrônico	
RADTEC	Modelo de banco de dados
SAF	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SARF	
SCCA	Modelo de banco de dados
SCP	Regras de negócio
SCV	Descrição de casos de uso Regras de negócio
Seg Ação	Descrição de casos de uso Modelo de banco de dados

<b>Nome do sistema</b>	<b>Documentação existente</b>
	Regras de negócio
SEG Web	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SELIN	
SGD	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SICAF	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SICAP	Descrição de casos de uso Modelo de banco de dados Regras de negócio
Siscorporativo	Descrição de casos de uso Documentação de regras de negócio
SISGESAC	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SISPA	Descrição de casos de uso Regras de negócio
SIP	
RADCOM	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SRH	Descrição de casos de uso Modelo de banco de dados Regras de negócio
REPUBL	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SRAPD	Descrição de casos de uso Modelo de banco de dados Regras de negócio
SEI	Descrição de casos de uso Modelo de banco de dados Regras de negócio
Conexão minicom	
Sítio Intranet	

Os sistemas foram categorizados em três grupos: sistemas que possuem o conjunto de documentações considerado a documentação mínima estabelecida pelo órgão, sistemas sem nenhuma documentação e sistemas que não possuem o conjunto de documentações mínimo mais possuem alguma documentação. É



observado por meio da Figura 10 que 50% dos sistemas possuem a documentação mínima, 26% dos sistemas não possuem documentação e 24% possuem alguma documentação.



**Figura 10:** Relatório de documentações presentes nos sistemas. Fonte: Autor.

A Tabela 27 representa a lista dos sistemas que possuem a documentação mínima estabelecida pelo *Ministério X*.

**Tabela 27:** Nome dos sistemas que possuem a documentação mínima. Fonte: Autor.

	Sistemas com documentação mínima
1	ASPAR
2	CPROD
3	Denúncia Web
4	DSCOM
5	OuvidoriaWeb
6	RADCOM
7	REPMBL
8	SAF
9	Seg Ação
10	SEG Web
11	SEI
12	SGD
13	SICAF
14	SICAP
15	SISGESAC
16	SRAPD
17	SRH

A Tabela 28 retrata a lista com o nome dos sistemas que não o conjunto de documentos considerados mínimos pelo órgão mas possuem alguma documentação.

**Tabela 28:** Relação de sistemas que possuem alguma documentação. Fonte: Autor.

	<b>Sistemas com alguma documentação</b>
1	CADSEI
2	OTRS
3	RADTEC
4	SCCA
5	SPC
6	SCV
7	SISPA
8	SISCORPORATIVO

A Tabela 29, mostra a lista com o nome dos sistemas que não possuem a documentação mínima estabelecida pelo *Ministério X*. Devido essa ausência de documentação não possível coletar as métricas desse objetivo de medição, por essa razão esses sistemas não contam na Tabela 30 que apresenta o resultado da coleta de métricas.

**Tabela 29:** Relação de nome de sistemas sem documentação. Fonte: Autor.

	<b>Sistemas sem nenhuma documentação</b>
1	CEL
8	Conexão minicom
2	GEO
3	GPD
4	Ponto Eletrônico
5	SARF
6	SELIN
7	SIP
9	Sítio Intranet

### **Questão 1: A documentação pode ser entendida pela equipe mantenedora?**

Os valores obtidos por meio das métricas para cada sistema estão representados na Tabela 30.

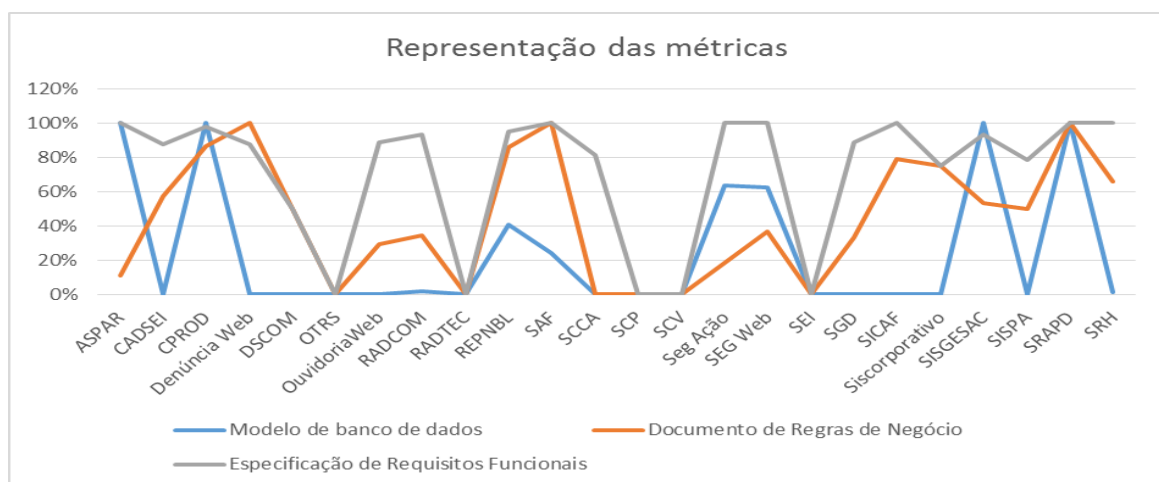
**Tabela 30:** Primeira questão do objetivo de medição 2. Fonte: Autor.

<b>Questão 1</b>	<b>M 1.1)</b> Porcentagem de elementos do modelo entidade relacionamento que estão documentados.	<b>M 1.2)</b> Indicador do nível de detalhamento dos requisitos	<b>M 1.3)</b> Indicador de qualidade dos requisitos	<b>M 1.4)</b> Porcentagem de descrição de requisitos funcionais que possuem descrição das regras de negócio	<b>M 1.5)</b> Indicador de facilidade de entendimento da especificação de requisitos funcionais
ASPAR	100%	Detalhado	Boa qualidade	11,32%	100%
CADSEI	0%	Detalhado	Pouca qualidade	57,14%	87,5%
CPROD	100%	Detalhado	Boa qualidade	86,67%	98,11%
Denúncia Web	0%	Detalhado	Pouca qualidade	100%	87,5%
DSCOM	0%	Pouco detalhado	Pouca qualidade	50%	50%
OTRS	0%	-	-	-	-
OuvidoriaWeb	0%	Detalhado	Pouca qualidade	29,62%	88,88%
RADCOM	1,75%	Detalhado	Boa qualidade	34,48%	93,54%
RADTEC	0%	-	-	-	-
REPMBL	40,68%	Detalhado	Boa qualidade	85,71%	95,23%
SAF	23,92%	Detalhado	Boa qualidade	100%	100%
SCCA	-	Detalhado	Boa qualidade	-	81,1%
SCP	-	-	-	-	-
SCV	0%	-	-	-	-
Seg Ação	63,41%	Bem detalhado	Boa qualidade	18,75%	100%
SEG Web	62,19%	Detalhado	Boa qualidade	36,84%	100%
SEI	0%	-	-	-	-
SGD	0%	Detalhado	Pouca qualidade	33,33%	88,88%
SICAF	0%	Detalhado	Boa qualidade	79,16%	100%
Siscorporativo	-	Pouco detalhado	Pouca qualidade	75%	75%
SISGESAC	100%	Detalhado	Boa qualidade	53,33%	93,33%
SISPA	-	Pouco detalhado	Boa qualidade	50%	78,57%
SRAPD	100%	Detalhado	Boa	100%	100%

Questão 1	M 1.1) Porcentagem de elementos do modelo entidade relacionamento que estão documentados.	M 1.2) Indicador do nível de detalhamento dos requisitos	M 1.3) Indicador de qualidade dos requisitos	M 1.4) Porcentagem de descrição de requisitos funcionais que possuem descrição das regras de negócio	M 1.5) Indicador de facilidade de entendimento da especificação de requisitos funcionais
			qualidade		
SRH	1,09%	Bem detalhado	Boa qualidade	65,62%	100%

Através dos resultados obtido pelas métricas presentes na Tabela 30 é destacado alguns sistemas que apesar de possuírem a documentação mínima estabelecida pelo *Ministério X* essa documentação não apresenta um bom resultado obtido por meio das métricas coletadas. Os sistemas que apresentam uma menor completude da documentação existentes são: CADSEI, DSCOM, SGD, SRH, SICAF, RADCOM e OuvidoriaWeb.

A Figura 11 representa os valores obtidos pelas métricas 1.1, 1.4 e 1.5. A linha azul do gráfico denominada modelo de banco de dados se refere aos valores da métrica 1.1 (Porcentagem de elementos do modelo entidade relacionamento que estão documentados). A linha laranja denominada documento de regras de negócio representa os valores da métrica 1.4 (Porcentagem de descrição de requisitos funcionais que possuem descrição das regras de negócio). Já a linha cinza representa os valores da métrica 1.5 (Indicador de facilidade de entendimento da especificação de requisitos funcionais). É observado uma grande variação dos valores obtidos por essas métricas.



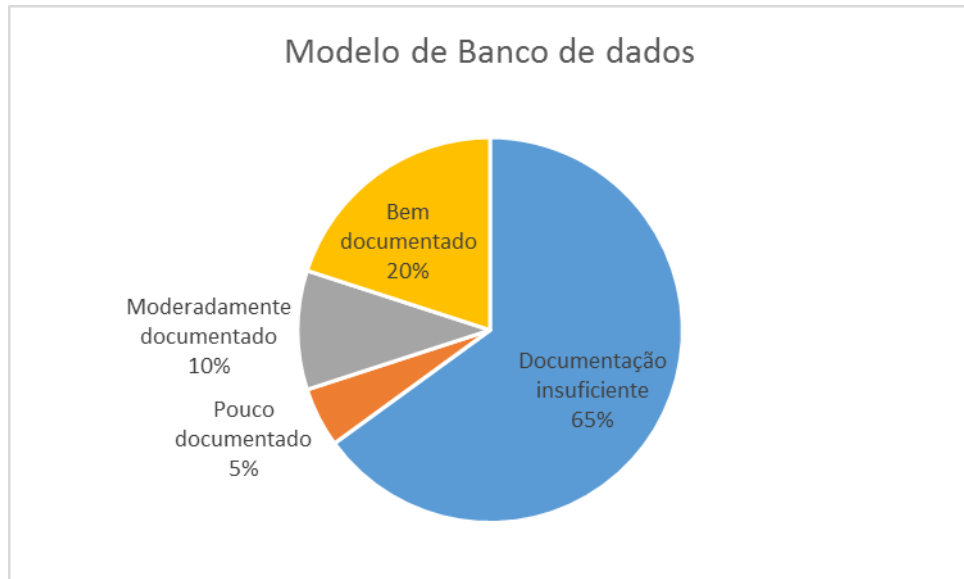
**Figura 11:** Representação dos valores das métricas para cada sistema. Fonte: Autor.

De acordo com Ramos (2004) quando o valor obtido pela métrica 1.1 (porcentagem de elementos do modelo entidade relacionamento que estão documentados) está entre 00% a 25% indica que a documentação é insuficiente, entre 25% e 50% que o sistema é pouco documentado, entre 50% a 80% moderadamente documentado, entre 80% a 100% bem documentado. A Tabela 31 relata esses valores e os sistemas pertencentes a cada classificação.

**Tabela 31:** Classificação do sistemas em relação a sua documentação. Fonte: Autor.

Sistemas	Intervalos	Interpretação
SRH SICAF SGD SEI SCV SAF RADTEC RADCOM ouvidoriaWeb OTRS DSCOM Denúncia Web CADSEI	00% a 25%	Documentação insuficiente
REPUBL	25% a 50%	Pouco documentado
SEG WEB Seg ação	50% a 80%	Moderadamente documentado
SRAPD SIGESAC CPROD ASPAR	80% a 100%	Bem documentado

Utilizando os dados apresentados na Tabela 31 pode-se observar que: 65% dos sistemas possuem documentação do modelo de banco de dados insuficiente, 10% dos sistemas possuem o modelo de banco de dados é pouco documentado, 5% moderadamente documentado, e 20% o modelo de banco de dados é bem documentado. Esses valores estão presentes na Figura 12.



**Figura 12:** Porcentagem de sistemas quanto a documentação do modelo de banco de dados. Fonte: Autor.

Quando o valor obtido pelas métricas 1.4 (porcentagem de descrição de requisitos funcionais que possuem descrição das regras de negócio) e 1.5 (indicador de facilidade de entendimento da especificação de requisitos funcionais), estão entre 00% a 25% indica que a documentação é insuficiente, entre 25% e 50% que o documento possui um baixo grau de descrição, entre 50% a 80% que a descrição é moderada, entre 80% a 100% que o documento possui um alto grau de descrição (Ramos, 2004).

A Tabela 32 relata esses valores e os sistemas pertencentes a cada classificação para a métricas 1.4 (porcentagem de descrição de requisitos funcionais que possuem descrição das regras de negócio).

**Tabela 32:** Classificação dos sistemas quanto aos seus requisitos funcionais. Fonte: Autor.

Sistemas	Intervalos	Interpretação
ASPAR Seg ação	00% a 25%	Documentação insuficiente
DSCOM OuvidoriaWeb RADCOM SEG WEB SGD	25% a 50%	Pouco documentado
CADSEI SICAF Siscorporativo SIGESAC SISPA SRH	50% a 80%	Moderadamente documentado
CPROD	80% a 100%	Bem documentado

Sistemas	Intervalos	Interpretação
Denúncia Web REPUBL SAF SRAPD		

Utilizando os dados apresentados na Tabela 32 pode-se observar que: 11% dos sistemas possuem descrição de requisito funcionais descritos nas regras de negócio de forma insuficientes, 25% dos sistemas são poucos documentado, 33% moderadamente documentado e 28% os requisitos são bem documentados. Esses valores estão representados na Figura 13.



**Figura 13:** Percentual de sistemas e completude da documentação de regras de negócio. Fonte: Autor.

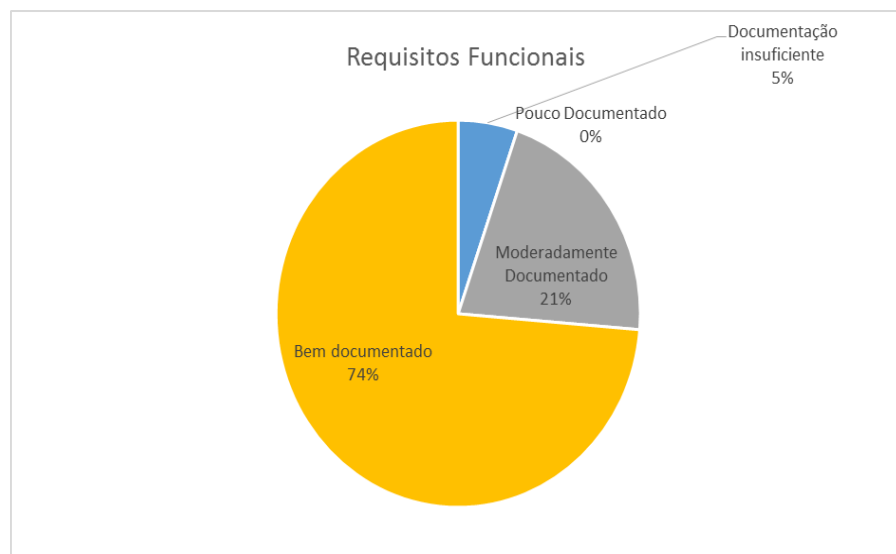
A Tabela 33 relata esses valores e os sistemas pertencentes a cada classificação para a métrica 1.5 indicador de facilidade de entendimento da especificação de requisitos funcionais.

**Tabela 33:** Relação de sistemas com a documentação de requisitos funcionais. Fonte: Autor.

Sistemas	Intervalos	Interpretação
ASPAR	00% a 25%	Documentação insuficiente
	25% a 50%	Pouco documentado
DSCOM Sisgesac SISPA SRH	50% a 80%	Moderadamente documentado
CADSEI	80% a 100%	Bem documentado

Sistemas	Intervalos	Interpretação
CPROD Denúncia Web ouvidoriaWeb RADCOM REPNBL SAF SCCA Seg ação Seg Web SGD SICAF Siscorporativo SRAP		

Utilizando os dados apresentados na Tabela 33 pode-se observar que: 5% dos sistemas possuem documentação de requisito funcionais insuficientes, 0% dos sistemas são poucos documentados, 21% moderadamente documentado e 74% os requisitos são bem documentados. Esses valores estão presentes na Figura 14.



**Figura 14:** Percentual de sistemas com os requisitos funcionais. Fonte: Autor.

A Tabela 34 representa as observações realizadas sobre o conteúdo da documentação de cada sistema. Essas observações foram realizadas durante a coleta das métricas e visam obter um melhor entendimento do conteúdo das documentações analisadas.



**Tabela 34:** Descrição de aspectos do conteúdo das documentações. Fonte: Autor.

<b>Sistemas</b>	<b>Modelo de banco de dados</b>	<b>Descrição de casos de uso</b>	<b>Documento de regra de negócio</b>
ASPAR	Bem documentado. Existe descrição para todas as entidades e atributos presentes no banco de dados.	Detalhado e de boa qualidade.	Poucos casos de uso descritos na documentação de regras de negócio. Considerada insuficiente.
CADSEI	Não existe descrição de entidades e atributos do modelo de dados.	Os requisitos são detalhados porém de baixa qualidade.	Porcentagem de casos de uso detalhado nas regras de negócio considerado insuficiente.
CEL	Não existe documentação.	Não existe documentação.	Não existe documentação.
Conexão minicom	Não existe documentação.	Não existe documentação.	Não existe documentação.
CPROD	Modelo de dados bem documentado todas as entidades e atributos possuem descrição	Descrição de casos de uso de boa qualidade e bem detalhado. Alto grau de compreensão.	Grande parte dos casos de uso possuem descrição na regras de negócio. Regras de negócio bem documentada.
Denúncia Web	Não existe descrição de entidades e atributos do modelo de dados.	Casos de uso pouco de detalhado e de baixa qualidade.	Regra de negócio bem documentada todos os casos de uso estão descritos.
DSCOM	Não existe descrição de entidades e atributos do modelo de dados.	Casos de uso pouco de detalhado e de baixa qualidade.	Moderadamente documentado.
GEO	Não existe documentação.	Não existe documentação.	Não existe documentação.
GPD	Não existe documentação.	Não existe documentação.	Não existe documentação.

Sistemas	Modelo de banco de dados	Descrição de casos de uso	Documento de regra de negócio
OTRS	Modelo de banco de dados sem descrição de entidades e atributos.	Não existe a documentação.	Não existe a documentação.
ouvidoriaWeb	Não existe descrição de entidades e atributos do modelo de dados.	Casos de uso pouco de detalhado e de baixa qualidade.	Regras de negócio pouco documentados.
Ponto Eletrônico	Não existe documentação.	Não existe documentação.	Não existe documentação.
RADCOM	Existe documentação de um pequeno número de atributos do modelo de dados.	Casos de uso bem documentados e de boa qualidade.	Regras de negócio pouco documentadas.
RADTEC	Não existe descrição de entidades e atributos do modelo de dados	Não existe documentação.	Não existe documentação.
REPNBL	O modelo de dados possui a documentação de entidades e atributos porém grande parte não estão documentados.	Descrição de casos de uso de boa qualidade e bem detalhado. alto grau de compreensão.	Grande parte dos casos de uso possuem descrição na regras de negócio. Regras de negócio bem documentada.
SAF	Documentação de atributos e entidades do modelo de dados considerada insuficiente.	Casos de uso detalhados e de boa qualidade.	Todos os casos de uso possuem descrição na documentação de regra de negócio.
SARF	Não existe	Não existe	Não existe

Sistemas	Modelo de banco de dados	Descrição de casos de uso	Documento de regra de negócio
	documentação.	documentação.	documentação.
SCCA	Não existe documentação.	Detalhado e de boa qualidade.	Não existe documentação.
SCP	Não existe descrição de entidades e atributos do modelo de dados.	Não existe descrição de requisitos.	As regras de negócio e o protótipos estão localizados no mesmo documento. Documento confuso sem organização e baixo entendimento.
SCV	Não existe descrição de entidades e atributos do modelo de dados.	Não existe documentação.	Não existe documentação.
Seg Ação	Modelo de banco de dados moderadamente documentado.	Casos de uso bem detalhado e de boa qualidade.	Documentação de regras de negócio insuficiente.
SEG Web	Documentação do modelo de dados considerada moderada.	Documentação de caso de uso considerado detalhada e de qualidade.	Documentação de regras de negócio pouco documentada.
SEI	Não existe descrição de entidades e atributos do modelo de dados	Não existe a documentação.	Não existe a documentação.
SELIN	Não existe documentação.	Não existe documentação.	Não existe documentação.
SGD	Não existe descrição de entidades e atributos do modelo de dados.	Casos de uso bem documentado.	Pouco documentado.
SICAF	Não existe	Casos de uso	Todos os casos de

Sistemas	Modelo de banco de dados	Descrição de casos de uso	Documento de regra de negócio
	descrição de entidades e atributos do modelo de dados.	bem documentado.	uso estão descritos nas regras de negócio.
SICAP	Não existe documentação.	Não existe documentação.	Não existe documentação.
SIP	Não existe documentação.	Não existe documentação.	Não existe documentação.
Siscorporativo	Não existe descrição de entidades e atributos do modelo de dados.	Descrição pouco detalhada e baixa qualidade.	Grande quantidade de requisitos descrito no documento de regras de negócio.
SISGESAC	Existe descrição para todas as entidades e atributos presentes no banco de dados.	Detalhado e de boa qualidade.	Porcentagem de casos de uso detalhado nas regras de negócio considerado insuficiente.
SISPA	Não existe nenhum tipo de documentação do modelo de dados do sistema.	Boa qualidade, casos de uso bem detalhados.	Moderadamente documentado.
Sítio Intranet	Não existe documentação.	Não existe documentação.	Não existe documentação.
SRAPD	Todos os atributos e entidades estão documentados.	Casos de uso detalhado e de boa qualidade.	Documentação de regra de negócio bem documentada. Todos os casos de uso possuem descrição nas regras de negócio.
SRH	Descrição de entidades e atributos insuficiente (pequeno número	Casos de uso bem detalhado e de boa qualidade.	Documentação de regras de negócio moderadamente documentada.

Sistemas	Modelo de banco de dados	Descrição de casos de uso	Documento de regra de negócio
	documentado).		

### 5.1.1.3 Objetivo de medição 3: análise da quantidade de requisições de manutenção corretiva

Nesta seção será apresentada a coleta de dados referente a análise de requisições de manutenção corretiva. Essa análise foi realizada verificando o número de requisições de manutenção corretiva e estudando o conteúdo dessas requisições. Isso foi realizado para verificar se o número total de requisições representava o número de erros encontrados nos sistemas.

Sistemas que possuem uma alta demanda requisições de manutenções corretivas, e essas demandas realmente se referem a erros presentes nos sistemas, são indicadores que a organização pode incluí-lo no novo contrato de manutenção. Pois o custo do sistema para a organização se equivale ao custo serviço prestado pela empresa mantenedora.

Caso o sistema tenha uma alto número de manutenção corretiva mas esse número de manutenções são referentes a outros aspectos relacionados a utilização do sistema não sendo erros do sistema. São indicativos que os problemas apresentados pelo sistema podem ser solucionados de outra maneira não tendo a necessidade de incluí-lo no novo contrato de manutenção. Pois esse grande número são referentes a outros aspectos do sistema que foram classificados de forma equivocada como manutenção corretiva.

A análise dos registros de manutenção corretiva foi realizada entre o período de 01 de abril de 2015 até 01 de setembro de 2015. Quando um sistema possuía menos de 5 registros esse período de análise foi prolongado para 28 de abril de 2014 até 01 de setembro de 2015 . A data de 28 de abril de 2014 foi escolhida pois essa é a data onde o OTRS foi implantado, ou seja a partir dessa data os registros de manutenção corretiva deixaram de ser realizados manualmente e começaram a ser feitos através da ferramenta.

Ampliando o tempo de análise do chamado de 28 de abril de 2014 até 01 de setembro de 2015, não foi encontrado nenhum sistema cujo o número de registro de manutenção corretiva dobrasse, ou pelo menos aumentasse de forma significativa

para justificar a análise em um período maior do que o período de análise utilizado por este trabalho. Dessa maneira foi mantido a análise dos registros de manutenção corretiva entre o período de 01 de abril de 2015 até 01 de setembro de 2015.

### **Questão 1: Qual o total de defeitos apresentados pelo sistema?**

Na lista de trinta e quatro sistemas que estão ativos no *Ministério X*, existem quatro sistemas, que estão ativos no portfólio de sistemas, mas não tinham a autorização do gestor para serem publicados na lista de serviços do ORTS. Como eles não foram publicados nessa lista de serviços, não foi possível o usuário registrar solicitações de manutenção corretiva pela ferramenta, e por isso eles não foram analisados nesta etapa do trabalho. Os sistemas não analisados são: SIP, SRAPD, GEO e segurança.

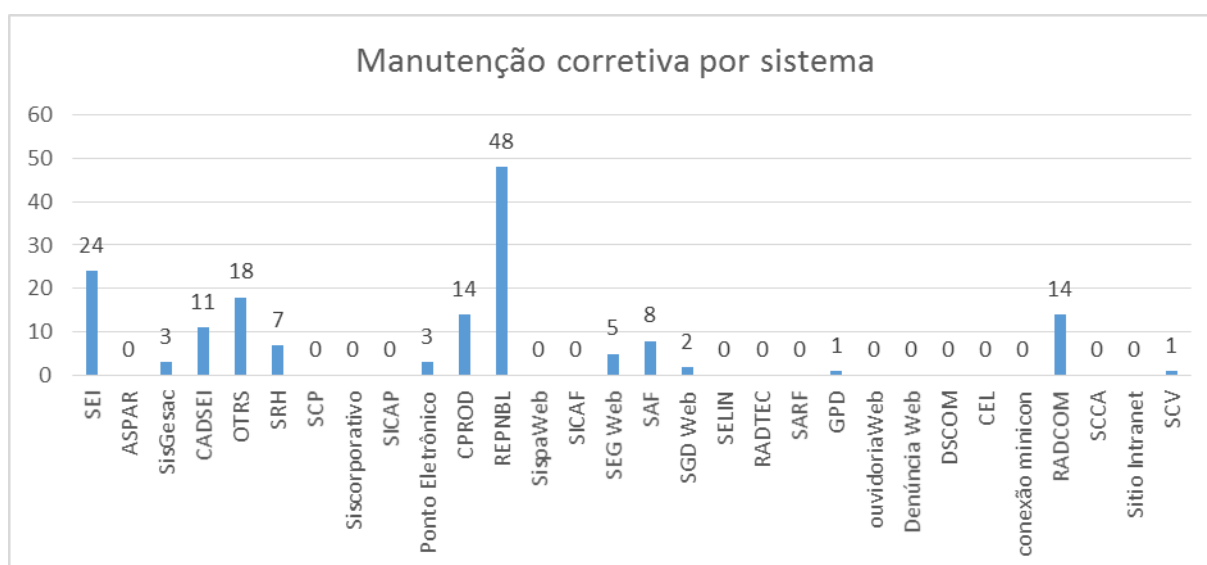
A Tabela 35 representa a quantidade de manutenções corretivas registradas na ferramenta OTRS, no período de 01 de abril de 2015 até 01 de setembro de 2015.

**Tabela 35:** Primeira questão do objetivo de medição 3. Fonte: Autor.

<b>Questão 1</b>	<b>M 1.1) Total de erros detectados</b>
ASPAR	0
CADSEI	11
CEL	0
Conexão minicom	0
CPROD	14
Denúncia Web	0
DSCOM	0
GPD	0
OTRS	18
ouvidoriaWeb	0
Ponto Eletrônico	3
RADCOM	14
RADTEC	0
REPUBL	48

Questão 1	M 1.1) Total de erros detectados
SAF	8
SARF	0
SCCA	0
SCP	0
SCV	1
SEG Web	5
SEI	24
SELIN	0
SGD	2
SICAF	0
SICAP	0
Siscorporativo	0
SISGESAC	3
SISPA	0
Sítio Intranet	0
SRH	7

Os valores referentes a quantidade de registros de manutenção corretiva encontrados no período de análise para cada sistema estão representados na Figura 15.



**Figura 15:** Número de manutenções corretivas por sistema. Fonte: Autor.

Utilizando os dados da Tabela 35, que corresponde a primeira questão desse objetivo de medição, foram classificados três grupos de sistemas o grupo de severidade alta, média, baixa e irrelevante. O grupo de nível de criticidade alto corresponde aos sistemas que possuem mais de 30 requisições de manutenção no período de análise, já nível de criticidade médio possuem entre 11 e 30, nível de criticidade baixo 5 a 10 e irrelevante são sistemas que possuem menos de 5 requisições.

A Tabela 36, relaciona o número de requisição de manutenção corretiva com os níveis de criticidade definidos.

**Tabela 36:** Valor de severidade de acordo com o número de manutenção corretiva. Fonte: Autor.

<b>Nível de Criticidade</b>	<b>Valor de referência</b>
Alta	$R > 30$
Média	$11 \leq R \leq 30$
Baixa	$5 \leq R \leq 10$
Irrelevante	$5 < R$

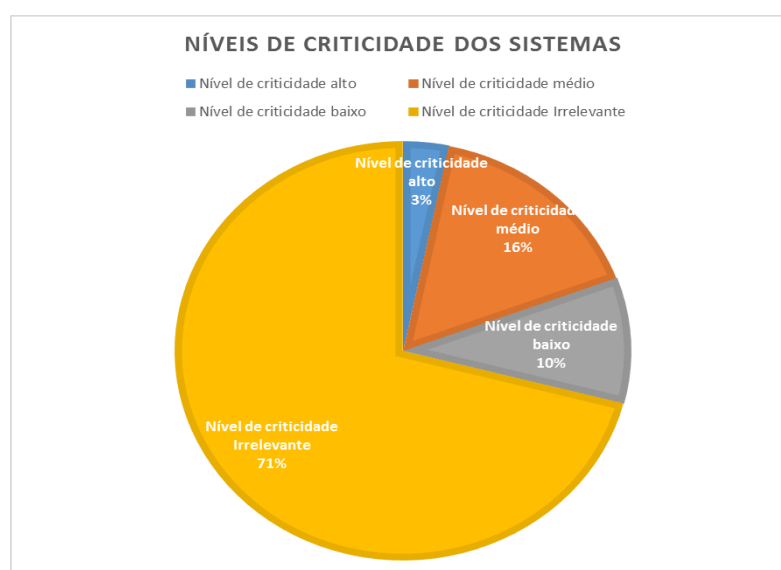
Os sistemas pertencentes a cada grupo de severidade definida por este trabalho estão listados na Tabela 37.

**Tabela 37:** Sistemas e s relação de severidades. Fonte: Autor.

<b>Nível de criticidade alto</b>	<b>Nível de criticidade médio</b>	<b>Nível de criticidade baixo</b>	<b>Nível de criticidade Irrelevante</b>
REPUBL	CRPOD SEI CADSEI OTRS RADCOM	SAF SEG Web SRH	ASPAR SIGESAC SPC Siscorporativo SICAP Ponto Eletrônico SISPA SICAF SGD SELIN RADTEC SARF GPD ouvidoriaWEB, DenúnciaWeb DSCOM CEL Conexão Minicom SCCA Sitio intranet SCV



Utilizando como base os dados da Tabela 37 são obtidos os seguintes dados: 71% dos sistemas possuem o nível de criticidade irrelevante, 10% possuem nível de criticidade baixo, 16% nível de criticidade médio e 3% nível de criticidade alto. Grande maioria dos sistemas possuem o nível de criticidade no aspecto de quantidade de manutenções corretivas irrelevante. Esses valores estão expostos no Figura 16.



**Figura 16:** Níveis de criticidade dos sistemas: Fonte: Autor.

### **Questão 2: Qual a classificação dos defeitos apresentados pelo sistema?**

A questão 2 deste objetivo de medição visa entender os erros apresentados nas requisições e verificar se realmente o que é registrado na ferramenta OTRS é realmente um erro ou uma falha do sistema ou outros aspectos que foram registrados como manutenção corretiva.

Realizando uma análise do histórico de cada registro de manutenção corretiva o número total de requisições foram divididos em diferentes aspectos. Sendo eles:

- Quantos registros eram realmente erros do sistema;
- Quantos registros eram erros mas estavam com o registro duplicado, ou seja eles descreviam um erro que já havia sido registrado antes e seria resolvido em um chamado registrado anteriormente;
- Requisições de manutenção corretiva classificadas com erro e foram resolvidas com essa classificação. Por exemplo, o chamado era um

problema de outra área como infraestrutura mas foram abertos como manutenção corretiva e resolvido como manutenção corretiva;

- Chamados cujo a descrição que o usuário informou não era suficiente para resolver a demanda, vale ressaltar que nesse caso o mantenedor do sistema entrou em contato com o usuário para obter esclarecimento sobre a demanda e não obteve resposta;
- Chamados feitos pelo usuário mas o erro apresentado pelo o mesmo não existia, nesses casos o próprio usuário não sabia utilizar o sistema e achava que o não funcionamento do sistema era falha do sistema e não uso indevido do mesmo.

A questão 2 desse objetivo de medição visa entender se o número de erros apresentado pelo sistema realmente são erros ou se são outros aspectos que estão sendo quantificados como erros do sistema. Dessa maneira esse objetivo faz uma triagem do número de defeitos registrados para obter o quanto desse número representa falhas ou defeitos do sistema.

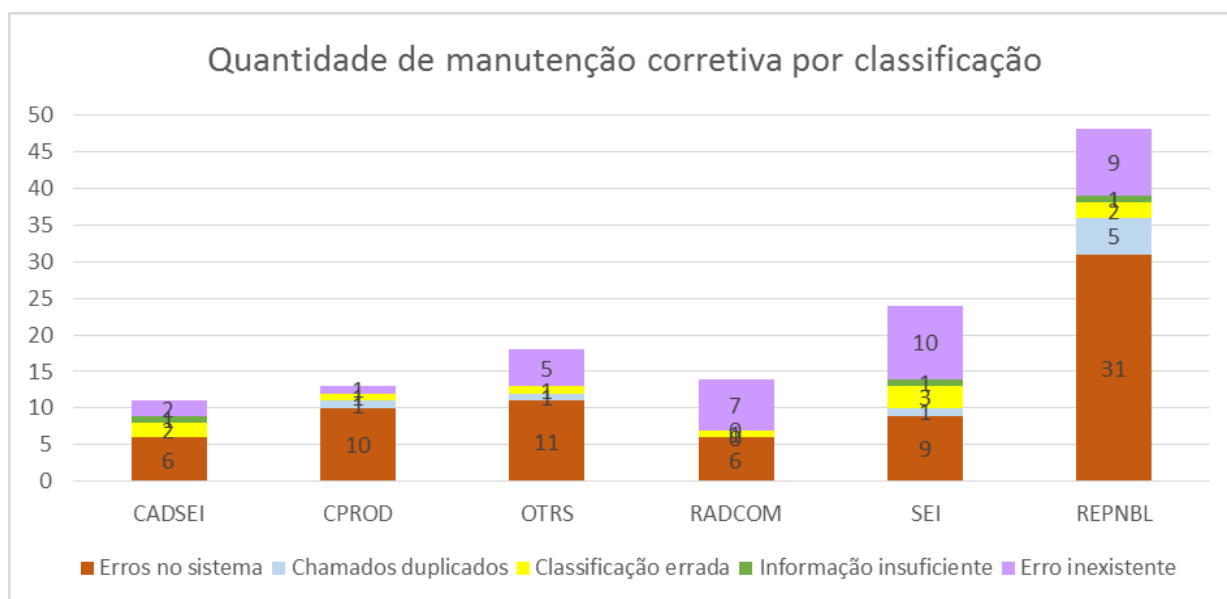
Os valores obtidos através das métricas selecionadas para a questão 2 estão descritos na Tabela 38.

**Tabela 38:** Segunda questão objetivo de medição 3. Fonte: Autor.

<b>Questão 2</b>	<b>M 2.1)</b> Total de erros apresentados pelas requisições de manutenção corretiva.	<b>M 2.1)</b> Total de chamados de manutenção corretiva que estão duplicados.	<b>M 2.2)</b> Total de chamados de manutenção corretiva que estão classificados errados.	<b>M 2.3)</b> Total de chamados de manutenção corretiva que não foram realizados pois a descrição do problema era inconclusiva	<b>M 2.4)</b> Total de chamados de manutenção corretiva que o erro apresentado na demanda não existe.
ASPAR	0	0	0	0	0
CADSEI	6	0	2	1	2
CEL	0	0	0	0	0
Conexão minicon	0	0	0	0	0
CPROD	10	1	1	0	1
Denúncia Web	0	0	0	0	0

<b>Questão 2</b>	<b>M 2.1) Total de erros apresentados pelas requisições de manutenção corretiva.</b>	<b>M 2.1) Total de chamados de manutenção corretiva que estão duplicados.</b>	<b>M 2.2) Total de chamados de manutenção corretiva que estão classificados errados.</b>	<b>M 2.3) Total de chamados de manutenção corretiva que não foram realizados pois a descrição do problema era inconclusiva</b>	<b>M 2.4) Total de chamados de manutenção corretiva que o erro apresentado na demanda não existe.</b>
DSCOM	0	0	0	0	0
GPD	0	0	0	0	0
OTRS	11	1	1	0	5
ouvidoriaWeb	0	0	0	0	0
Ponto Eletrônico	0	0	1	0	2
RADCOM	6	0	1	0	7
RADTEC	0	0	0	0	0
REPNBL	31	5	2	1	9
SAF	7	0	0	0	1
SARF	0	0	0	0	0
SCCA	0	0	0	0	0
SCP	0	0	0	0	0
SCV	0	0	0	0	1
SEG Web	3	0	0	0	2
SEI	9	1	3	1	10
SELIN	0	0	0	0	0
SGD	1	0	0	0	1
SICAF	0	0	0	0	0
SICAP	0	0	0	0	0
Siscorporativo	0	0	0	0	0
SISGESAC	3	0	0	0	0
SISPA	0	0	0	0	0
Sitio Intranet	0	0	0	0	0
SRH	2	0	1	0	4

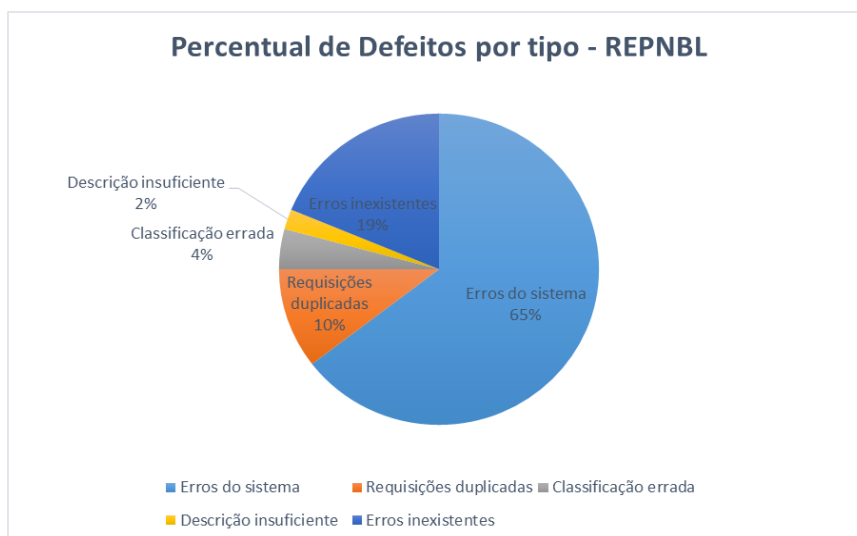
A Figura 17 retrata as cinco classificações feitas nos chamados de manutenção corretiva obtidas pelas métricas presentes na Tabela 38. A Figura 17 considera somente os sistemas classificados com nível de criticidade alto e médio, utilizando os números presentes nesta mesma tabela para realizar uma melhor visualização dos dados.



**Figura 17:** Representação das classificações das manutenções corretivas. Fonte: Autor.

A seguir será apresentado de forma detalhada as classificações dos registros de manutenção corretiva dos sistemas de nível de criticidade alto e médio, sendo eles: REPUBL, CRPOD, SEI, CADSEI, OTRS e RADCOM. Reportando também alguns defeitos mais recorrentes identificados durante a coleta das métricas presentes na Tabela 38 .

No sistema REPUBL 64,58% do número de erros encontrados são erros dos sistema, já os outros 18,75% o erro era inexistente, 10,41% eram requisições duplicadas, 4,16% requisições que tiveram sua classificação errada e 2,08% eram requisições cujo a descrição era insuficiente para resolver a demanda. Esses valores estão demonstrados na Figura 18.

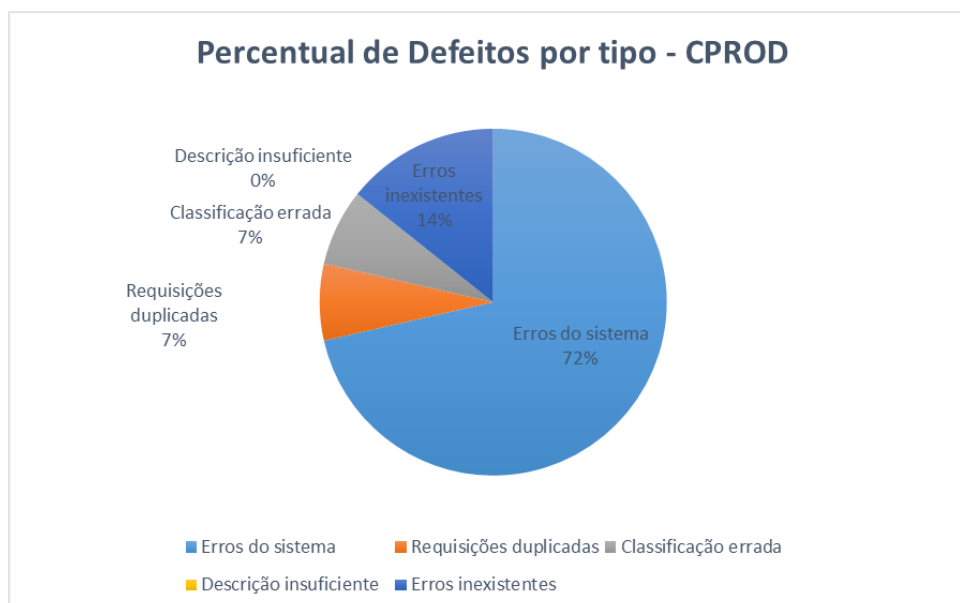


**Figura 18:** Percentual de defeitos do sistema REPABL. Fonte: Autor.

A maioria do número de requisições de manutenção corretiva apresentado no REPABL são realmente defeitos. Alguns desses defeitos recorrentes do sistema são apresentado a seguir:

- Arquivos duplicados no banco de dados que impedem o upload de novos arquivos;
- O sistema não permite a exportação de dados;
- Erros ao cadastrar relatórios;
- O sistema não mostra a última versão do projeto salvo no mesmo;
- Erro na busca de um projeto a partir da sua data de submissão;
- O sistema passa a exigir campos obrigatórios que não existiam antes.

O sistema CPDOD possui 72% do número de erros encontrados são erros dos sistema, já os outros 14% eram erros inexistentes, 7% requisições duplicadas, 7% requisições que tiveram sua classificação errada e 0% eram requisições cujo a descrição era insuficiente para resolver a demanda. Esses valores estão demonstrados na Figura 19.

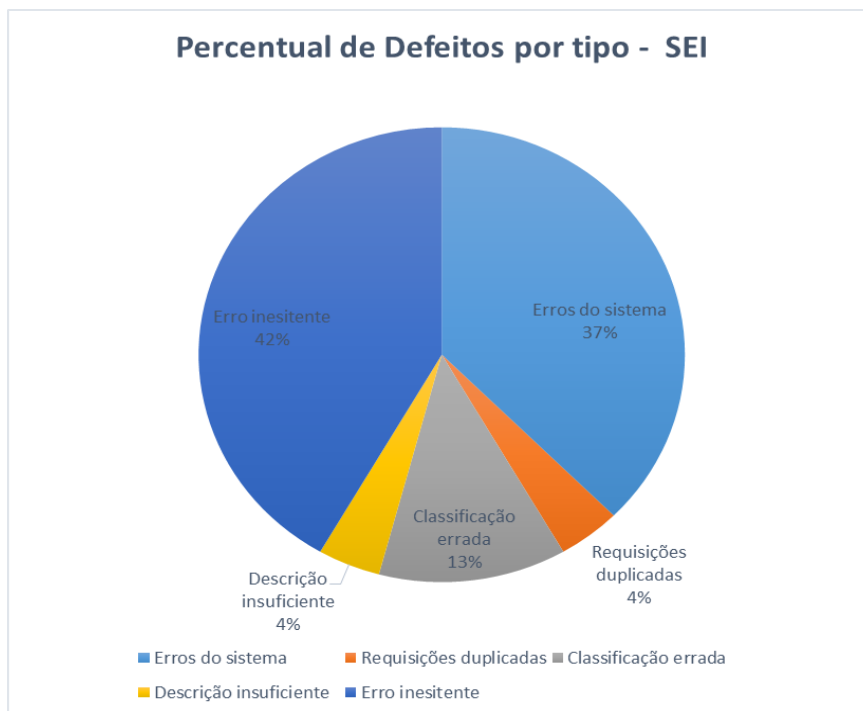


**Figura 19:** Percentual de defeitos por tipo do sistema CPROD. Fonte: Autor.

No sistema CPROD grande percentual do número das requisições são defeitos. Alguns desses defeitos recorrentes do sistema são apresentado a seguir:

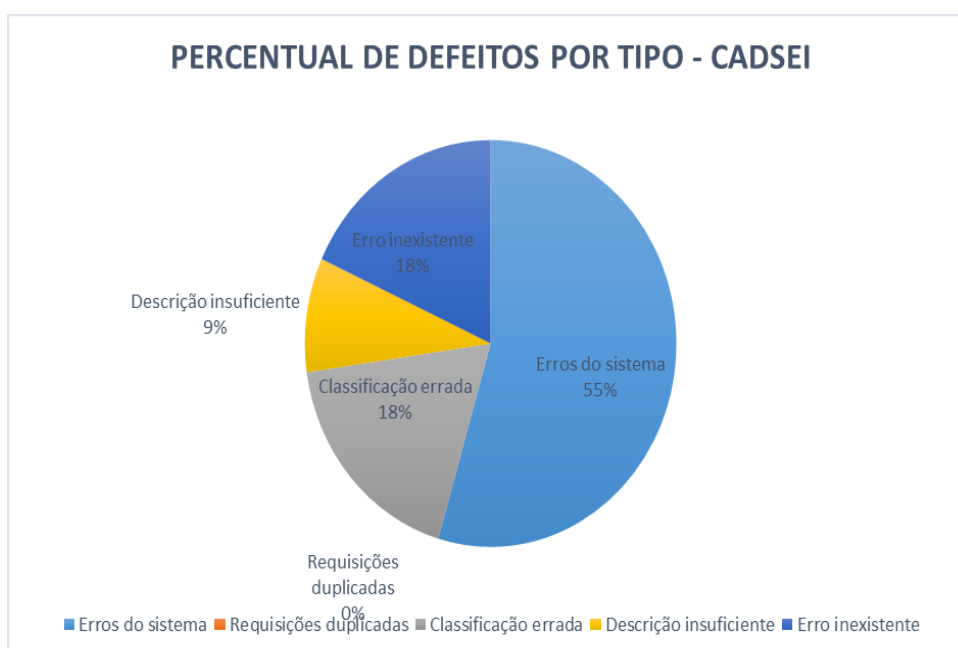
- O sistema não retorna os resultados da busca realizada;
- A janela do sistema não carrega o que é solicitado;
- A aba das informações gerais desaparecem quando é realizado uma consulta;
- O sistema não realiza a consulta dos processos.

O sistema SEI está classificado com criticidade média apresentando 24 requisições de manutenção corretiva no período de análise. A maioria dessas requisições não eram realmente defeitos do sistema e sim erros inexistentes. De um total de 37,5% do número de erros encontrados são realmente erros dos sistema, já os outros 41,66% o erro era inexistente, 0% eram requisições duplicadas, 12,5% requisições que tiveram sua classificação errada e 4,16% eram requisições cujo a descrição era insuficiente para resolver a demanda. Esses valores estão demonstrados na Figura 20.



**Figura 20:** Percentual de defeitos por tipo do sistema SEI. Fonte: Autor.

O sistema CADSEI apresenta 54,54% do número de erros encontrados são realmente erros dos sistema, já os outros 18,18% o erro é inexistente, 0% eram requisições duplicadas, 18,18% requisições que tiveram sua classificação errada e 9,09% eram requisições cujo a descrição era insuficiente para resolver a demanda. Esses valores estão demonstrados na Figura 21.



**Figura 21:** Percentual de defeitos por tipo do sistema CADSEI. Fonte: Autor.

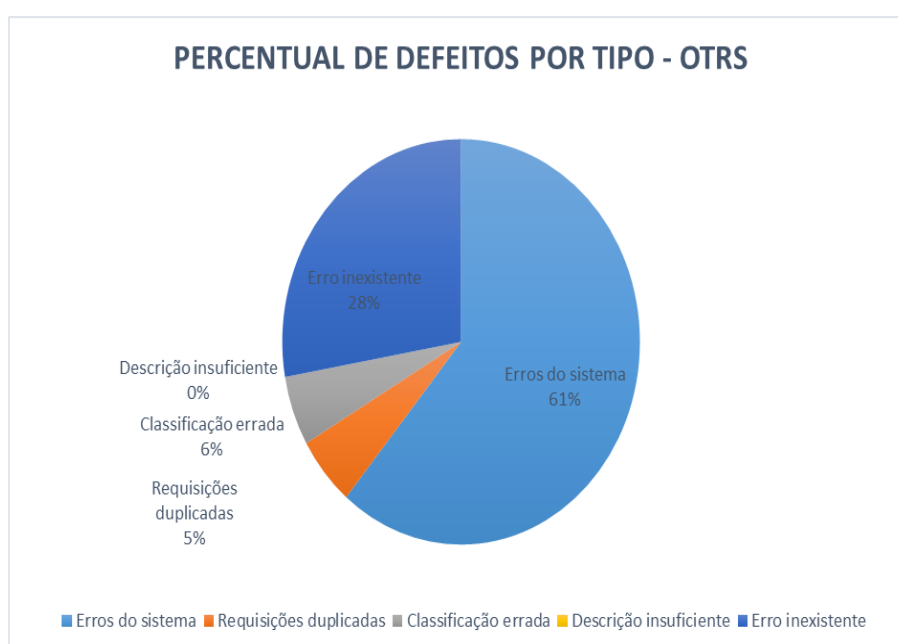
No sistema CADSEI grande número das requisições são defeitos, porém o sistema apresentou somente 11 requisições de manutenção corretiva no período analisado. Alguns desses defeitos recorrentes do sistema são apresentado a seguir:

- Erro na comunicação entre o CADSEI e o corporativo;
- Cadastro de cpf duplicado entre o CADSEI e o corporativo;
- Falha ao enviar arquivos;
- O índice de identificação aparece incorreto.

No sistema OTRS grande número das requisições são defeitos. Alguns desses defeitos recorrentes do sistema são apresentado a seguir:

- O sistema não estava enviando o e-mail para o responsável da demanda;
- Técnicos não conseguem reclassificar um chamado;
- O sistema deixa de permitir o acesso as FAQ já cadastradas;
- Os chamados não mudam de estado após as atualizações.

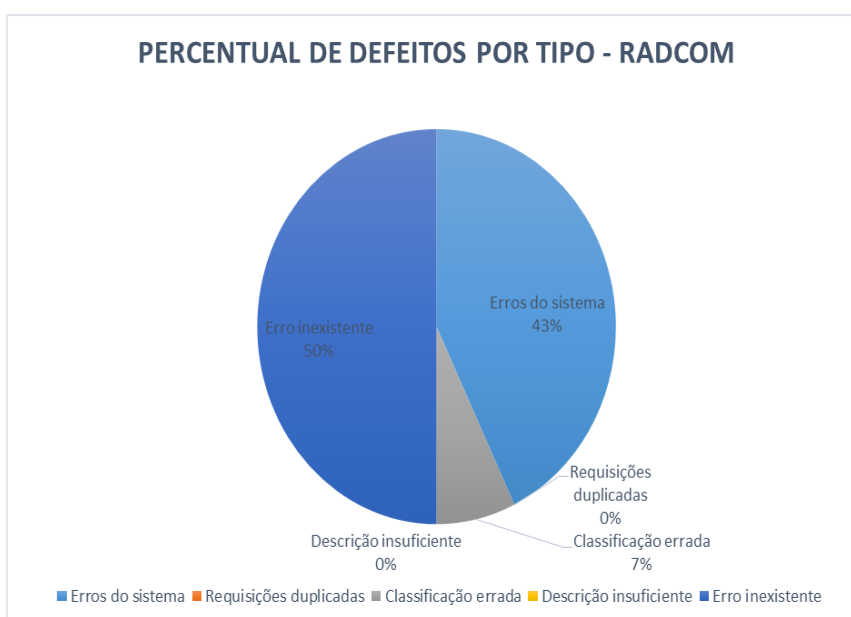
O OTRS 61,11% do número de erros encontrados são realmente erros dos sistema, já os outros 27,77% o erro é inexistente, 5,55% eram requisições duplicadas, 5,55% requisições que tiveram sua classificação errada e 9,09% eram requisições cujo a descrição era insuficiente para resolver a demanda. Esses valores estão demonstrados na Figura 22.



**Figura 22:** Percentual de defeitos por tipo do sistema OTRS. Fonte: Autor.



No sistema RADCOM o percentual de defeitos que são erros do sistema e que o erro apresentado era inexistente se assemelha. Das requisições de manutenção corretiva 42,85% são realmente erros dos sistema, já os outros 50% o problema apresentado pelo usuário não existia, 0% eram requisições duplicadas, 7,14% requisições que tiveram sua classificação errada e 0% eram requisições cujo a descrição era insuficiente para resolver a demanda. Esses valores estão demonstrados na Figura 23.



**Figura 23:** Percentual de defeitos por tipo do sistema RADCOM. Fonte: Autor.

Através da análise dos dados já apresentados e utilizando os sistemas classificados nesse trabalho como sistemas com nível de criticidade alto e sistemas nível de criticidade médio são obtidos os seguintes resultados: no sistema REPUBL 64,58% das requisições são realmente erros, o CPROD apresenta mais 72% das requisições como erros, o SEI 37%, o CADSEI apresenta 54,54%, o OTRS apresenta 61,11% e o RADCOM 43%.

## 6 COMPARAÇÃO E ANÁLISE DOS DADOS

Este Capítulo traz uma análise dos resultados já apresentados. Utilizando como base as métricas obtidas no Capítulo 5 para criar um conjunto de indicadores que foram comparados com o propósito de observar suas relações nos sistemas. Logo após foi feita uma análise do número de manutenção corretiva durante os meses. Por último apresenta-se a matriz de decisão para softwares legados.

### 6.1 COMPARAÇÃO ENTRE OS DADOS

Para realizar a comparação entre os dados foram utilizados três valores como referências: a métrica RCI que expressa um valor de qualidade para os softwares legados, a média das métricas de documentação (que estão presentes no item 5.1.1.2 desse trabalho) é o número de manutenções corretivas registradas na ferramenta OTRS.

A métrica RCI representa o valor para a qualidade dos sistemas, ela foi calculada utilizando um plug-in da ferramenta sonarQube. O RCI utiliza valores das métricas já calculadas pelo sonarQube para o projeto e geram um valor com escala de 0 a 1 equivale a qualidade interna do código. Quanto mais perto de 0 menor a qualidade interna do código e quanto mais perto de 1 melhor a qualidade interna do código.

O RCI é calculado utilizando os valores das métricas referentes a erros do sistema e o tamanho do em linhas de código. Cada erro apresentado pelo sistema possuem um peso que pode ser configurado na ferramenta de acordo com as necessidades da organização (SONAR, 2015).

Neste trabalho erros com gravidade muito alta possuem peso igual a 10, gravidade alta peso igual a 5, gravidade média igual a 3, e gravidade baixa 1 esses valores estão representados na Tabela 39: **Gravidade dos erros e pesos utilizados.** Fonte: Autor. Tabela 39.

**Tabela 39:** Gravidade dos erros e pesos utilizados. Fonte: Autor.

Gravidade dos erros	Peso
Erros com gravidade muito alta	10
Erros com gravidade alta	5
Erros com gravidade média	3

Gravidade dos erros	Peso
Erros com gravidade baixa	1

A fórmula para o cálculo do RCI está representada na Equação 1:

**Equação 1:** Formula do RCI. Fonte: (Adaptado, Sonar, 2015).

$$RCI = \frac{1 - ((EGMA * 10) + (EGA * 3) + (EGM * 3) + (EGB * 1))}{LCO}$$

**Onde:**

EGMA = Erros com gravidade muito alta;

EGA = Erros com gravidade alta;

EGM = Erros com gravidade média;

EGB = Erros com gravidade baixa;

LCO = Tamanho do sistema em linhas de código.

Os valores referentes aos erros e tamanho do sistema em linha de código foram calculados de forma automatizada pela ferramenta sonarQube, e estão presentes na seção 5.1 desse trabalho. Dentro do objetivo de medição 1 (avaliação da qualidade interna do código), já a quantidade de erros presentes em cada sistema se encontra na questão 4 (qual a quantidade de erros encontrados no código?). Os valores referentes a tamanho do sistema em linha de código estão na questão 1 (qual o tamanho do sistema?). Com a junção das métricas dessas duas questões é realizado o cálculo do RCI.

Como o cálculo da métrica RCI foi realizado através da ferramenta SonarQube não foi possível obter o valor para os sistemas: SARF, GPD, CEL, OTRS e sitio intranet. Para os sistemas SARF, GPD, CEL, e OTRS a ferramenta não possui suporte para as linguagens em que eles foram construídos, o sistema sitio intranet não possui código fonte para análise estática em seu repositório. A justificativa desses sistemas não terem métricas coletadas através da ferramenta apresenta-se de forma mais detalhada na seção 5.2 desse trabalho, no objetivo de medição 1 (avaliação da qualidade do código do sistema legado).

Para obter um valor referente a documentação de cada sistema foi calculado a média dos valores já obtidos pelas métricas de documentação já calculadas, que estão presentes na seção 5.1.1.2. Ressalta-se que os valores das métricas se encontram em porcentagem, é para o cálculo da média todos foram divididos por 100 para a obter de um valor numérico. Os sistemas que não possuíam a documentação tem o valor para o cálculo igual a zero.

A fórmula para o cálculo do índice de documentação está representado na Equação 2.

**Equação 2:** Formula do índice de documentação. Fonte: Autor.

$$ID = \frac{M1.1+M1.4+M1.5}{3}$$

**Onde:**

ID = Índice de documentação;

M1.1 = Valor da métrica 1.1, objetivo de medição 2;

M1.4 = Valor da métrica 1.4, objetivo de medição 2;

M1.5 = Valor da métrica 1.5, objetivo de medição 2.

O valor referente a quantidade de manutenções corretivas corresponde ao número de manutenções corretivas registradas no OTRS no período de análise, que se encontra na seção 5.1.1.3. Os sistemas em que os registros de manutenção corretiva não foram analisados tem o número de manutenção igual a zero.

Na Tabela 40 possui os valores referentes a métrica RCI, ao índice de documentação e ao número de requisições de manutenção corretiva.

**Tabela 40:** Relação de sistemas com RCI, índice de documentação e número de manutenções.  
Fonte: Autor.

Sistemas	RCI	Índice de documentação	Número de manutenções corretivas.
ASPAR	0,111	0,704	0
CADSEI	0	0,482	11
CEL	-	0	0
Conexão minicon	0,317	0	0

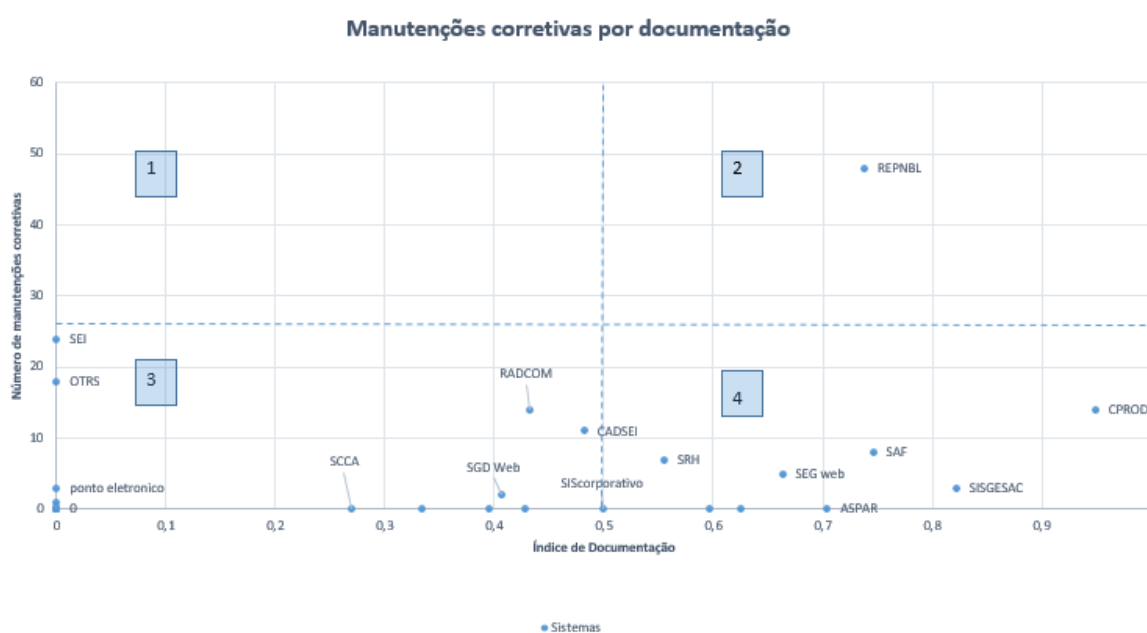
Sistemas	RCI	Índice de documentação	Número de manutenções corretivas.
CPROD	0	0,949	14
DenúnciaWeb	0,089	0,625	0
DSCOM	0	0,333	0
GEO	0,246	0	0
GPD	-	0	0
OTRS	-	0	18
OuvidoriaWeb	0	0,395	0
Ponto Eletrônico	0,054	0	3
RADCOM	0	0,432	14
RADTEC	0,952	0	0
REPUBL	0	0,738	48
SAF	0	0,746	8
SARF	-	0	0
SCCA	0,111	0,270	0
SCP	0	0	0
SCV	0,004	0	1
Seg Ação	0,090	0,607	0
SEG Web	0,154	0,663	5
SEI	0	0	24
SELIN	0	0	0
SGD	0,138	0,407	2
SICAF	0	0,597	0
SICAP	0,155	0	0
SIP	0	0	0
Siscorporativo	0,272	0,500	0
SISGESAC	0	0,822	3
SISPA	0	0,428	0
Sitio Intranet	-	0	0
SRAPD	0	1	0
SRH	0	0,555	7

### 6.1.1 Manutenção corretiva e documentação

Nessa seção é realizada a relação entres os valores numéricos presentes na Tabela 40, utilizando somente os valores das colunas referentes ao número de manutenções corretivas e ao índice de documentação.

Os valores plotados na Figura 24 são a associação entre a quantidade de manutenção corretiva do sistema e seu índice de documentação, cada ponto desse gráfico representa um sistema. Como alguns sistemas possuem os valores do número de manutenções corretivas e índice de documentação iguais os pontos se sobrepõem no gráfico não permitindo a visualização de todos os sistemas.

A Figura 24 foi dividida em quatro quadrantes para representar características dos sistemas presentes em cada quadrante.



**Figura 24:** Divisão de quadrantes para análise. Fonte: Autor.

O primeiro quadrante corresponde aos sistemas que possuem mais de 25 requisições de manutenções corretivas e menos de 0,5 de índice de documentação. O segundo quadrante são sistemas mais de 25 requisições de manutenções corretivas e mais de 0,5 de índice de documentação. O terceiro são sistemas com

menos de 25 manutenções corretivas e menos de 0,5 de índice de documentação, já o quarto os sistemas possuem menos de 25 manutenções corretivas e mais de 0,5 de índice de documentação.

Sistemas pertencentes ao quadrante número um são sistemas com um alto número de manutenção corretiva e um baixo índice de documentação, no quadrante número dois são sistemas com alto número de manutenção corretiva e alto índice de documentação, no número três sistemas com baixo número de manutenção corretiva e baixo índice de documentação, já o quadrante número quatro são sistemas com baixo número de manutenção corretiva e alto índice de documentação. A Tabela 41 contém esses valores.

**Tabela 41:**Relação entre o número de manutenções corretivas e índice de documentação. Fonte: Autor.

Quadrante	Valor de referência	Interpretação
1	MC > 25 & ID < 0,5	Alto número de manutenções corretivas e baixo índice de documentação.
2	MC > 25 & ID > 0,5	Alto número de manutenções corretivas e alto índice de documentação.
3	MC < 25 & ID < 0,5	Baixo número de manutenções corretivas e baixo índice de documentação.
4	MC < 25 & ID > 0,5	Baixo número de manutenções corretivas e alto índice de documentação.

A Tabela 42 possui todos os sistemas analisados e seus respectivos quadrantes.

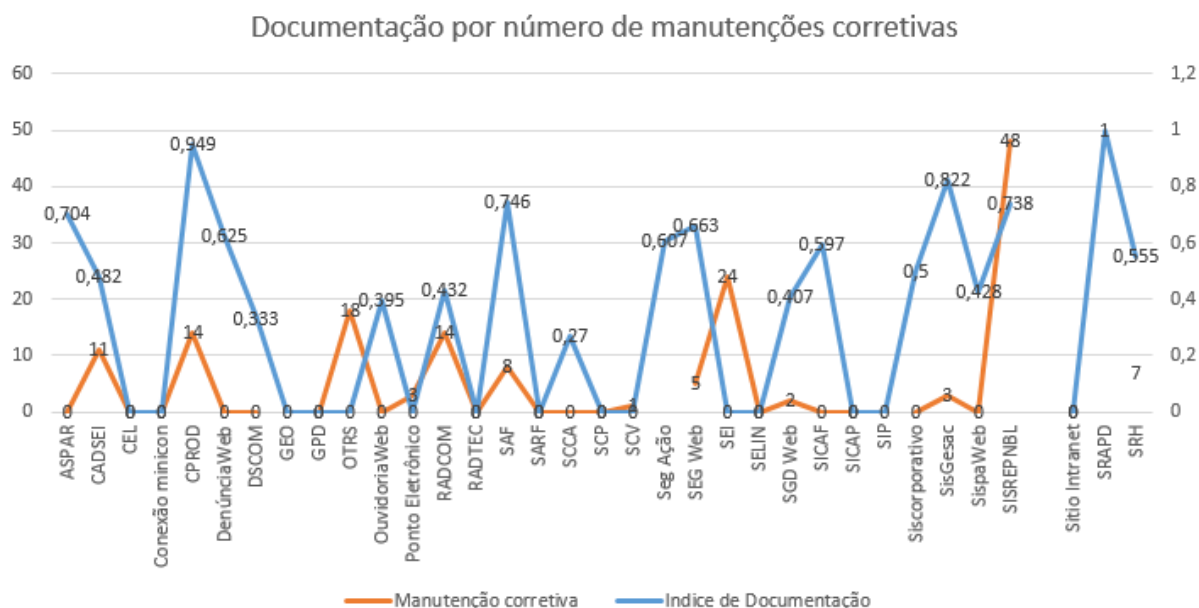
**Tabela 42:** Relação dos sistemas pertencentes a cada quadrante. Fonte: Autor.

Quadrante	Sistemas	Interpretação
1	-	Alto número de manutenções corretivas e baixo índice de documentação.
2	REPNBL	Alto número de manutenções corretivas e

Quadrante	Sistemas	Interpretação
		alto índice de documentação.
3	CADSEI CEL Conexão Minicon DSCOM GEO GPD OTRS Ouvidoria Web Ponto Eletrônico RADCOM RADTEC SARF SCCA SCP SCV SEI SELIN SICAP SIP Siscorporativo SipaWeb Sitio Intranet	Baixo número de manutenções corretivas e baixo índice de documentação.
4	CPROD Denúncia Web SAF Seg Ação Seg Web SICAF SIGESAC SRAPD SRH	Baixo número de manutenções corretivas e alto índice de documentação.

A Figura 25 possui duas linhas principais que representam o número de manutenção corretiva dos sistemas e a azul o índice de documentação. Ele permite uma melhor representação dos dois valores obtidos para cada sistema. O eixo vertical possui todos os sistemas já o eixo horizontal apresenta os valores das variáveis analisadas para cada sistema.





**Figura 25:** Representação da documentação e do número de manutenções corretivas. Fonte: Autor.

A hipótese H1 definida nesse trabalho no Capítulo 4 seção 4.1.1.2, afirma que:

*H1. O volume de manutenção corretiva realizadas nos sistemas legados é inversamente proporcional a documentação;*

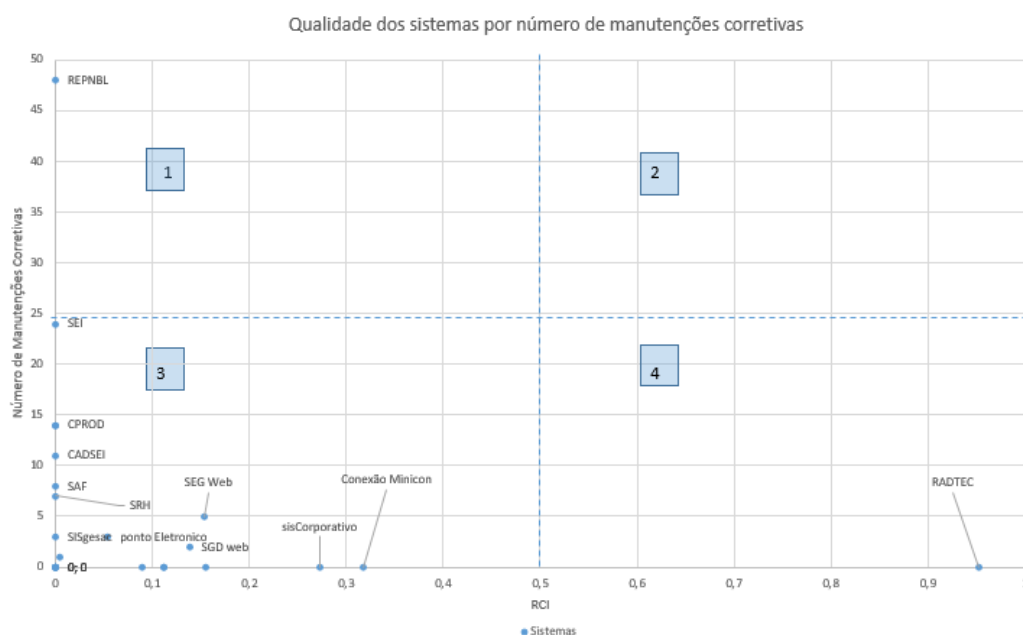
De acordo com a hipótese definida sistemas com um alto índice de documentação possuem um baixo número de manutenções corretivas. Como de acordo com a Tabela 42, não existe nenhum sistema com um baixo índice de documentação e uma grande quantidade de manutenção corretiva e o sistema REPNBL possui um alto índice de documentação e um alto número de manutenção corretiva pode-se observar que essas duas variáveis não se relacionam.

Na Tabela 42, encontra-se 21 sistemas que possuem um baixo índice de documentação e baixo número de manutenção corretiva. Provando que essa hipótese foi refutada, pois de acordo com os dados apresentados as variáveis número de manutenção corretiva e índice de documentação não são inversamente proporcionais.

### 6.1.2 Manutenção corretiva e qualidade

Essa seção analisa a relação entre duas variáveis distintas. Os números de manutenção corretiva registrados para um sistema e a qualidade interna do código.

Para obter essa relação é utilizado os valores das variáveis que estão presentes na Tabela 40 do item 6.1. Essas variáveis foram expressas na Figura 26.



**Figura 26:** Representação dos quadrantes. Fonte: Autor.

Com o propósito de observar o relacionamento entre as variáveis a Figura 26 foi dividida em quatro quadrantes. No primeiro quadrante encontra-se sistemas que possuem mais de 25 requisições de manutenções corretivas e menos de 0,5 de RCI, o segundo são sistemas mais de 25 requisições de manutenções corretivas e mais de 0,5 de RCI, o terceiro possuem menos de 25 manutenções corretivas e menos de 0,5 de RCI, já o quarto possuem menos de 25 manutenções corretivas e mais de 0,5 de RCI.

Sistemas pertencentes ao quadrante número um são sistemas com um alto número de manutenção corretiva e uma baixa qualidade, no quadrante número dois são sistemas com alto número de manutenção corretiva e alta qualidade, no número três sistemas com baixo número de manutenção corretiva e baixa qualidade, já o quadrante número quatro são sistemas com baixo número de manutenção corretiva e baixa qualidade. Esses valores estão presentes na Tabela 43.

**Tabela 43:** Interpretação dos valores pertencentes a cada quadrante. Fonte: Autor.

Quadrante	Valor de referência	Interpretação
1	MC > 25 & RCI < 0,5	Alto número de manutenções corretivas e alta qualidade.
2	MC > 25 & RCI > 0,5	Alto número de manutenções corretivas e alta qualidade.
3	MC < 25 & RCI < 0,5	Baixo número de manutenções corretivas e baixa qualidade.
4	MC < 25 & RCI > 0,5	Baixo número de manutenções corretivas e alta qualidade.

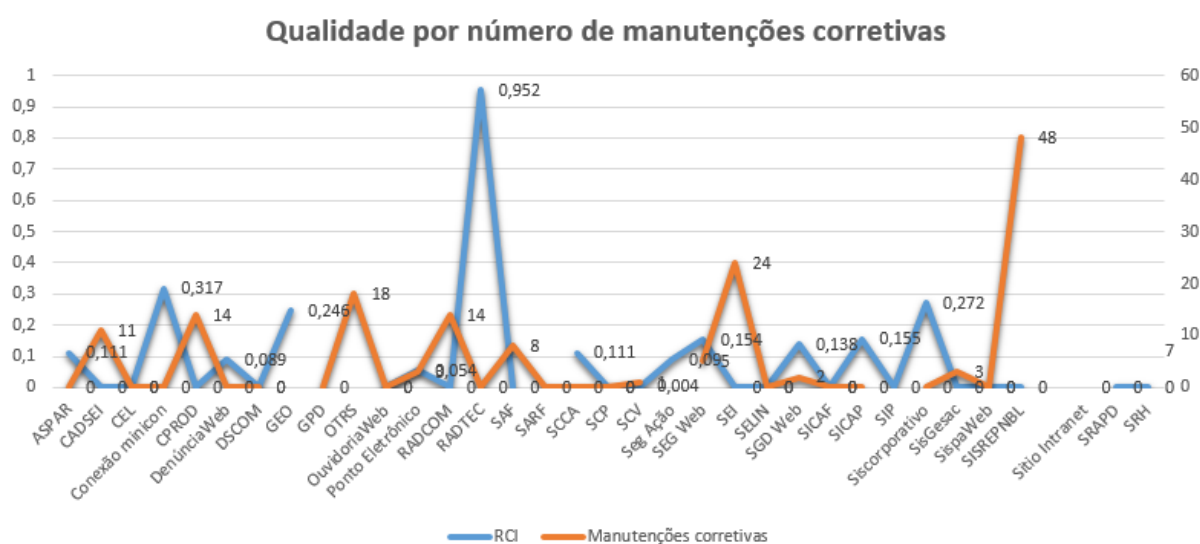
Os sistemas pertencentes a cada quadrante estão representados na Tabela 44.

**Tabela 44:** Relação de sistemas pertencentes a cada quadrante. Fonte: Autor.

Quadrante	Sistemas	Interpretação
1	REPNBL	Alto número de manutenções corretivas e baixa qualidade.
2	-	Alto número de manutenções corretivas e alta qualidade.
3	ASPAR CADSEI Conexão Minicon CPROD Denúncia Web DSCOM GEO OuvidoriaWeb Ponto Eletrônico RADCOM SAF SCCA SCV Seg ação Seg Web SEI SELIN SGD SICAF SICAP SIP	Baixo número de manutenções corretivas e baixa qualidade.

Quadrante	Sistemas	Interpretação
	SipaWeb Siscorporativo SIGESAC SRAPD SRH	
4	RADTEC	Baixo número de manutenções corretivas e alta qualidade.

Como os sistemas da Figura 26 possuem os valores de número de manutenção corretiva e RCI idênticos ficaram vários pontos na mesma localização no gráfico, o que não permite a representação de todos os sistemas. Para melhor expressar esses valores foi utilizada a Figura 27.



**Figura 27:** Representação da qualidade e os número de manutenção corretiva dos sistemas. Fonte: Autor.

A hipótese H2 definida nesse trabalho no Capítulo 4 seção 4.1.2, afirma que:

*H2. Os sistemas que possuem uma alto número de requisições de manutenção corretiva possuem uma baixa qualidade interna do código;*

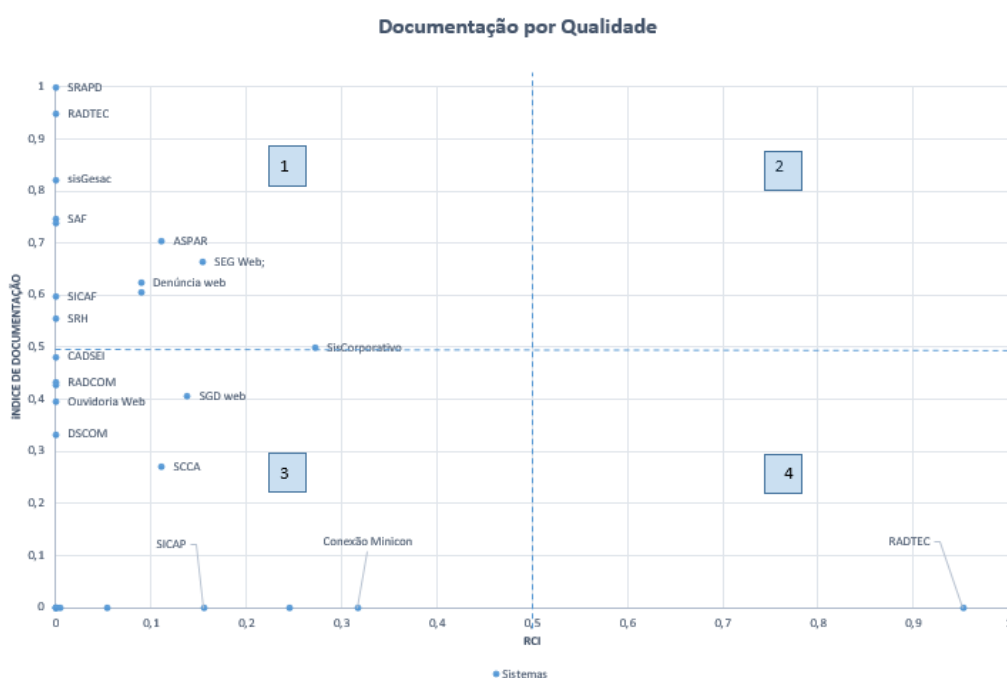
A partir da Tabela 44 pode-se observar que existe somente um sistema com baixa qualidade e elevado número de manutenção corretiva, o sistema REP-NBL. Como somente esse sistema possui essa característica e 26 sistemas possuem um baixo número de manutenção corretiva e baixa qualidade isso demonstrando que a

hipótese H2 foi refutada, pois não foi possível verificar a relação entre os valores das variáveis RCI e número de manutenções corretivas.

### 6.1.3 Qualidade e Documentação

Essa seção avalia a relação entre a qualidade do código e a documentação. Utilizando duas variáveis presentes na Tabela 40, que se localiza no item 6.1 desse trabalho, que são: o RCI e o índice de documentação.

A Figura 28 expressa a relação entre esses dois valores. O eixo vertical possui os valores referentes ao RCI, já o eixo horizontal são os valores referentes ao índice de documentação dos sistemas.



**Figura 28:** Divisão do quadrantes. Fonte: Autor.

Para melhor interpretação dos valores a Figura 28 foi dividida em quatro quadrantes. O primeiro corresponde a sistemas que possuem o índice de documentação maior que 0,5 e menos de 0,5 de RCI o segundo são sistemas com índice de documentação maior que 0,5 e RCI maior que 0,5, o terceiro possui índice de documentação maior que 0,5 e RCI que 0,5, já o quarto possuem índice de documentação menor que 0,5 e RCI maior que 0,5.

Sistemas pertencentes ao quadrante número um são sistemas com um alto índice de documentação e baixo RCI, no quadrante número dois são sistemas com alto índice de documentação e alto RCI, no número três sistemas com baixo índice de documentação e baixo RCI, já o quadrante número quatro são sistemas com baixo índice de documentação e alto RCI. Esses valores estão representados na Tabela 45.

**Tabela 45:** Interpretação dos valores do quadrantes. Fonte: Autor.

Quadrante	Valor de referência	Interpretação
1	$RCI > 0,5 \ \& \ ID < 0,5$	Alto índice de documentação e baixa qualidade.
2	$RCI > 0,5 \ \& \ ID > 0,5$	Alto índice de documentação e alta qualidade.
3	$RCI < 0,5 \ \& \ ID < 0,5$	Baixo índice de documentação e baixa qualidade.
4	$RCI < 0,5 \ \& \ ID > 0,5$	Baixo índice de documentação e alta qualidade.

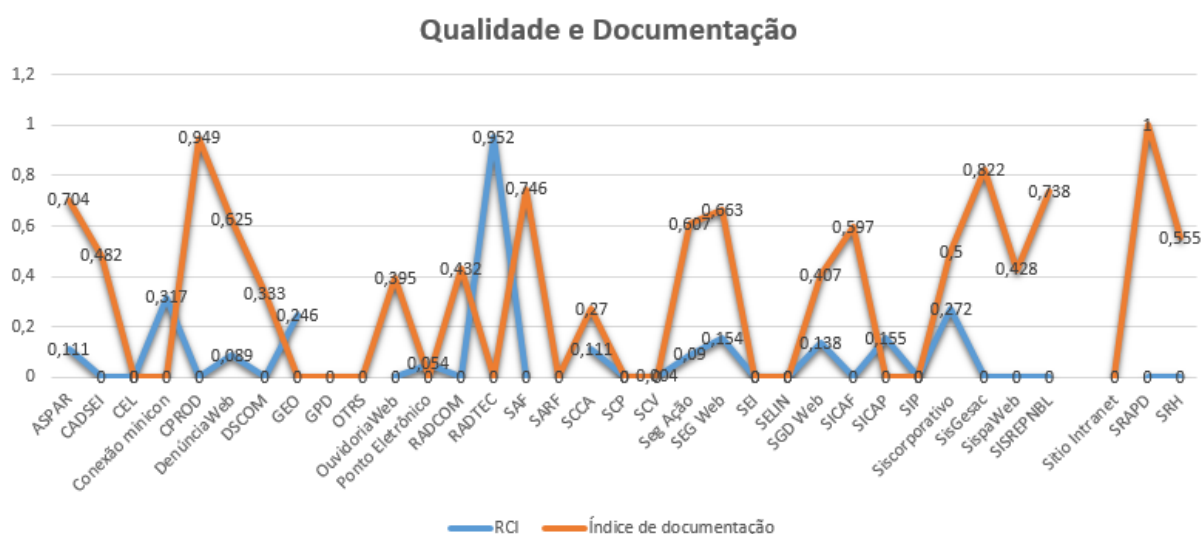
A relação dos sistemas analisados com os quadrantes ao qual cada um pertence se encontra na Tabela 46.

**Tabela 46:** Sistemas pertencentes a cada quadrante. Fonte: Autor.

Quadrante	Sistemas	Interpretação
1	ASPAR CPROD Denúncia Web REPNBL SAF Seg Ação Seg Web SICAF Siscorporativo SISGESAC SRAPD SRH	Alto índice de documentação e baixa qualidade.
2	-	Alto índice de documentação e

Quadrante	Sistemas	Interpretação
		alta qualidade.
3	CADSEI DSCOM ConexãoMinicon GEO Ponto Eletrônico OuvidoriaWeb RADCOM SCCA SGD SCP SCV SEI SELIN SICAP SIP SISPA	Baixo índice de documentação e baixa qualidade.
4	RADTEC	Baixo índice de documentação e alta qualidade.

A Figura 29 demonstra os valores das duas variáveis analisadas nessa seção. É observado que não existe nenhum sistema em que o índice de documentação e a qualidade se assemelham. Sistemas como por exemplo o CPROD possuem um alto índice de documentação e uma baixa qualidade interna do código, já o sistema RADTEC possui um baixo índice de documentação e uma alta qualidade interna.



**Figura 29:** Representação dos valores de qualidade e documentação dos sistemas. Fonte: Autor.

A hipótese H3 definida nesse trabalho no Capítulo 4 seção 4.1.1.2, afirma que:

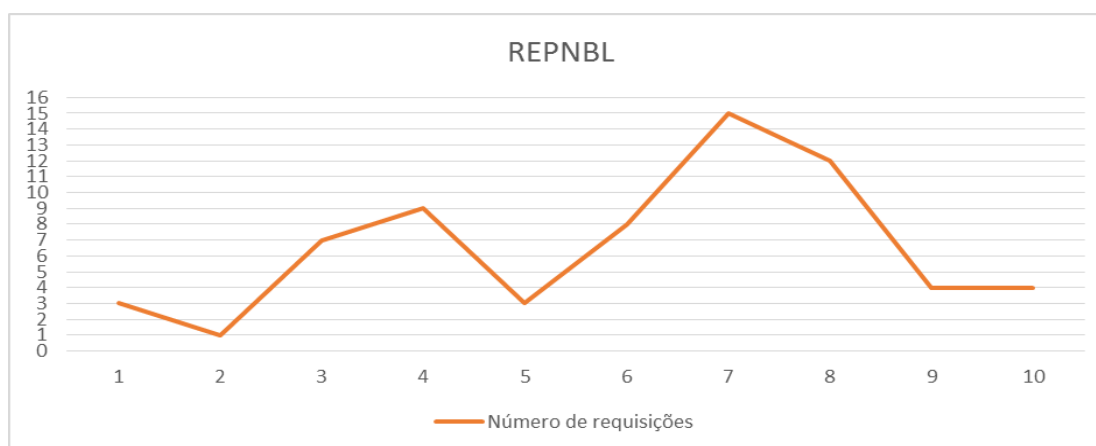
*H3. Os sistema que possuem baixa qualidade possuem um baixo nível de documentação*

Relacionando o índice de documentação com a qualidade interna do código pode-se observar (através da dos dados da Tabela 46) que 16 sistemas possuem baixa qualidade e baixo índice de documentação provamos que a hipótese *H3: Os sistema que possuem baixa qualidade interna do código possuem um baixo nível de documentação*, é verdadeira.

## 6.2 REGISTROS DE MANUTENÇÃO CORRETIVA

Esta seção apresenta a análise da evolução registros de requisição de manutenção corretiva presentes na ferramenta OTRS somente para os sistemas que no item 5.1.2.3 deste trabalho foram classificados como sistemas de nível de criticidade alto e médio. O período de análise corresponde aos meses de janeiro a outubro de 2015.

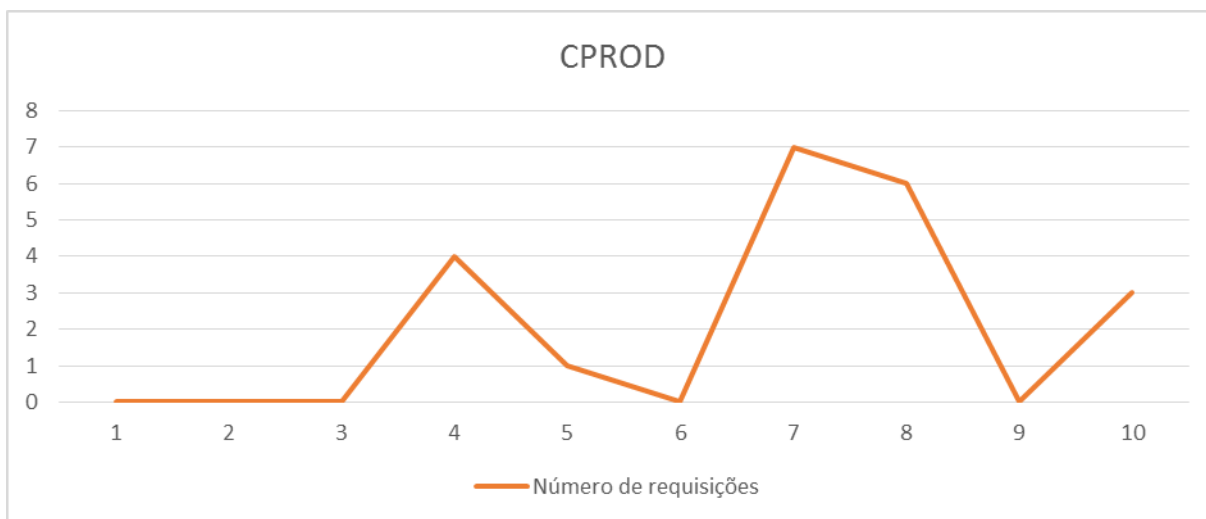
A Figura 30 representa a evolução dos registros das requisições de manutenção corretiva para o sistema REPABL. Pode-se observar que o período em que o sistema apresentou o maior número de registros correspondem ao mês 7 (julho) com 15 requisições e 8 (agosto) com 12 requisições.



**Figura 30:** Registro de manutenção corretiva sistema REPABL. Fonte: Autor.

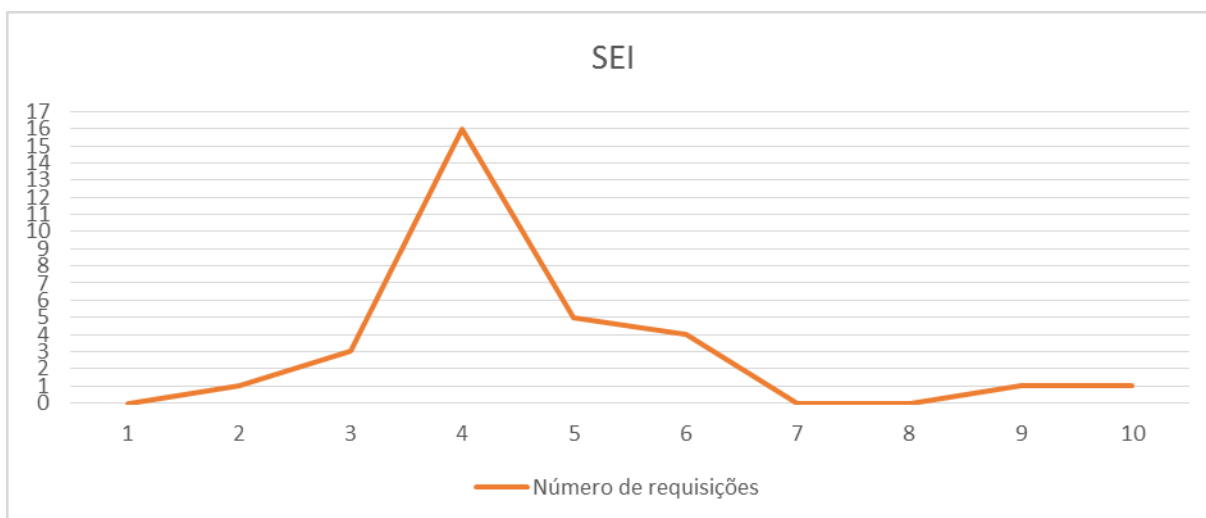


A Figura 31 representa a evolução dos registros das requisições de manutenção corretiva para o sistema CPROD. Através do gráfico observa-se que as quantidades de registros oscilaram durante os meses analisados. O período em que o sistema apresentou o maior número de manutenção corretiva correspondem ao mês 7 (julho) com 7 requisições e 8 (agosto) com 6 requisições.



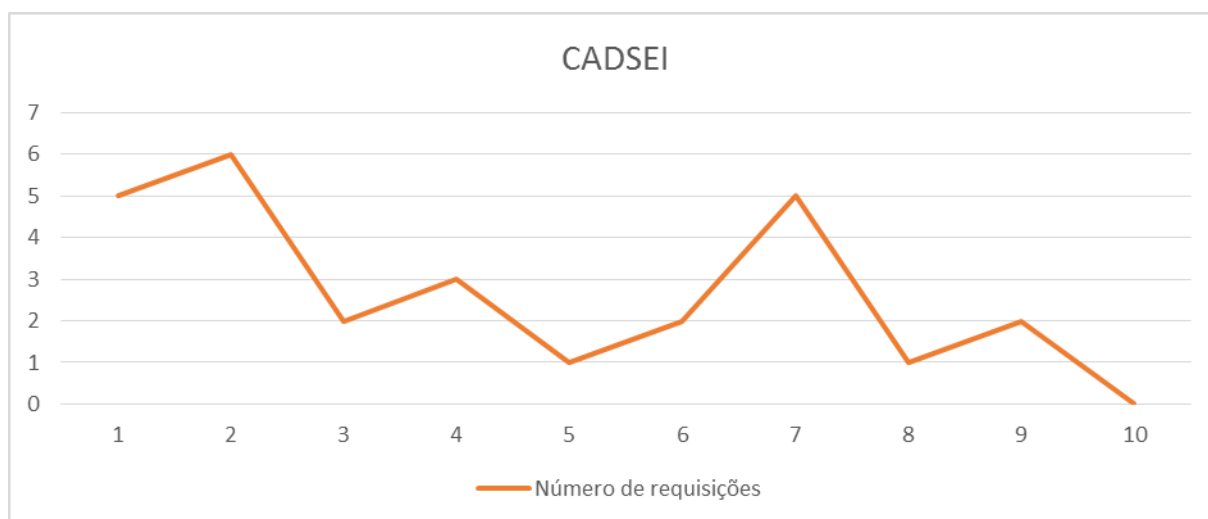
**Figura 31:** Registro de manutenção corretiva sistema CPROD. Fonte: Autor.

A Figura 32 expressa a evolução dos registros das requisições de manutenção corretiva para o sistema SEI. O período em que o sistema apresentou o maior número de manutenção corretiva correspondem ao mês 4 (março) com 16 após esse mês pode-se observar uma queda representativa dos registros.



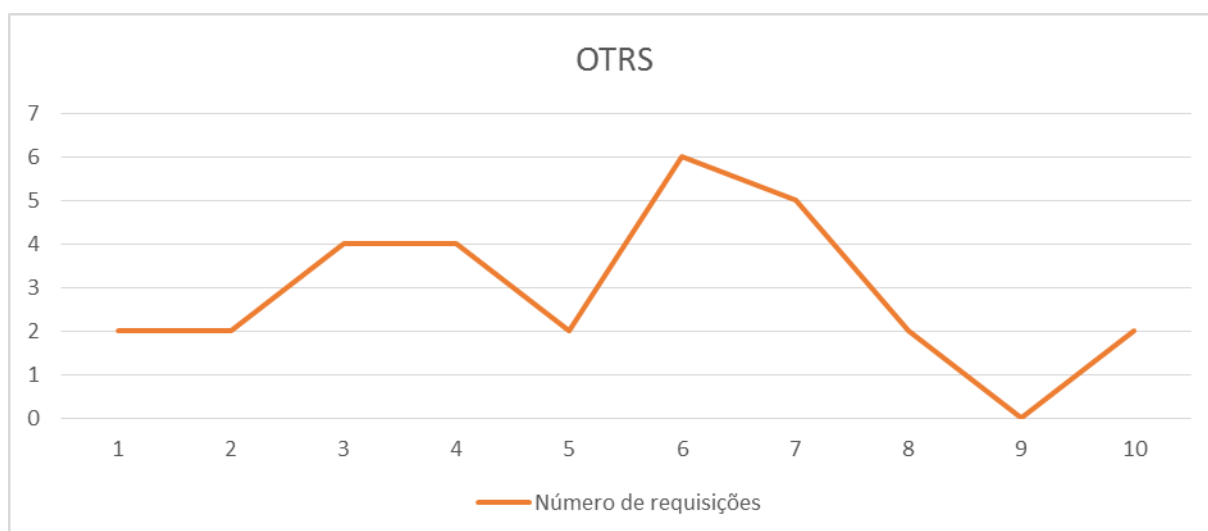
**Figura 32:** Registro de manutenção corretiva sistema SEI. Fonte: Autor.

A Figura 33 mostra a evolução dos registros das requisições de manutenção corretiva para o sistema CADSEI. Através do gráfico observa-se que as quantidades de registros oscilaram durante os meses analisados. O período em que o sistema apresentou o maior número de manutenção corretiva correspondem aos meses de 1 (janeiro) com 5 requisições, 2 (fevereiro) com 6 e 7 (julho) com 5.



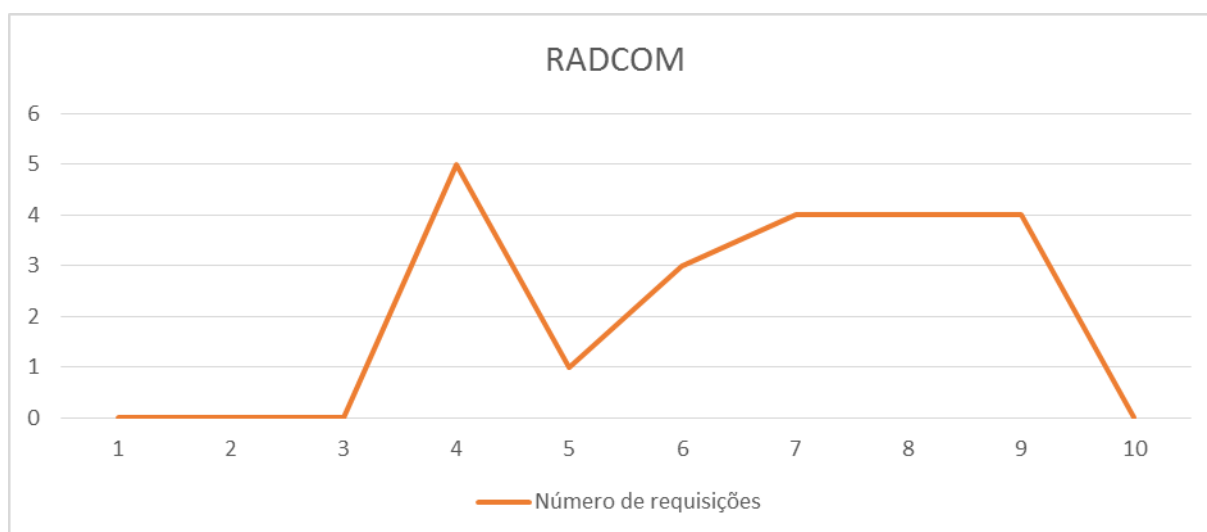
**Figura 33:** Registros de manutenção corretiva sistema CADSEI. Fonte: Autor.

A Figura 34 mostra os registros de manutenção corretiva durante os meses do ano de 2015 para o sistema OTRS. Através do gráfico observa-se que existem apenas dois meses em que houve uma grande quantidade de registros, o mês 6 (junho) com 6 requisições e 7 (julho) com 7.



**Figura 34:** Registro de manutenção corretiva sistema OTRS. Fonte: Autor.

A Figura 35 representa a evolução dos registros das requisições de manutenção corretiva para o sistema RADCOM. Através do gráfico observa-se que as quantidades de registros oscilaram durante os meses analisados. Nos primeiros meses do ano de 2015, não havia registros, tendo uma alta no mês 4 (março) com 5 requisições, já os meses 7 (julho) 8 (agosto) e 9 (setembro) obtiveram 4 registros de manutenção corretiva.



**Figura 35:** Registro de manutenção corretiva sistema RADCOM. Fonte: Autor.

### 6.3 MATRIZ DE DECISÃO PARA SOFTWARES LEGADOS

Softwares legados são sistemas que para serem mantido pela organização em que estão inseridos geram um alto custo. Existem decisões que podem ser tomadas sobre eles para a redução desse custos. Essas decisões são baseadas em dois aspectos presentes no sistema (SOMERVILLE, 2011). Nesse trabalho essas decisões são baseadas na qualidade interna do código por meio do RCI e a criticidade do sistema para a organização.

O RCI é uma métrica calculada pela ferramenta sonarQube, a especificação dessa métrica está presente no item 6.1. A criticidade do sistema para a organização é retirada a partir dos valores de criticidade ou urgência do serviço, esses valores foram definidos pelo *Ministério X*.

Todas as manutenções realizadas nos softwares legados do Ministério X são caracterizadas como serviços prestados. Quando um serviço é classificado com criticidade alta significa que eles são amplamente utilizados no Ministério X e são fundamentais para o trabalho dos usuários, sua interrupção causa um grande prejuízo para a organização. Serviços de criticidade média devem estar disponíveis mas sua interrupção é tolerada. Já serviços com criticidade baixa são atendidos sob demanda rotineira.

Neste trabalho os sistemas que possuem criticidade alta (para prestação de serviço de manutenção corretiva) tem o valor igual 3, criticidade média igual 2 e criticidade baixa igual a 1. Atualmente nenhum dos sistemas ativos no Ministério X possuem criticidade baixa, todos possuem criticidade média ou alta.

O documento de registro das criticidades atribuídas para cada sistema, não continha todos os sistemas que foram definidos como ativos pelo órgão. Dos 34 sistemas ativos o documento continha apenas 27, os outros sete que não contam nesse documento são: Denúncia Web, GEO, RADCOM, SCV, SEG ação, SIP, SRAPD. E devido à ausência desses sistemas eles não foram analisados nesta etapa do trabalho.

Os sistemas SARF, GPD, CEL, OTS e sitio intranet não foram analisados nesta etapa do trabalho pois esses sistemas não possuem o valor do RCI (justificativa da ausência desse valor presente no Capítulo 5 seção 5.2).

A Tabela 47 mostra a relação entre criticidade do serviço e os sistemas pertencentes a cada criticidade assim como o valor atribuído.

**Tabela 47:** Valor da criticidade dos sistemas. Fonte: Autor.

<b>Criticidade dos Serviços</b>	<b>Sistemas</b>	<b>Valor</b>
Baixa	-	1
Média	ASPAR CEL Conexão Minicon CPROD Denúncia Web DSCOM GPD RADTEC SAF SARF SCCA SCP SEG Web	2

<b>Criticidade dos Serviços</b>	<b>Sistemas</b>	<b>Valor</b>
	SELIN SGD SICAFI SICAP SISGESAC SISPA	
Alta	CADSEI OTRS OuvidoriaWeb Ponto Eletrônico REPUBL SEI SisCorporativo Sitio Intranet SRH	3

Os valores referentes ao RCI e a criticidade dos sistemas que foram analisados estão presentes na Tabela 48.

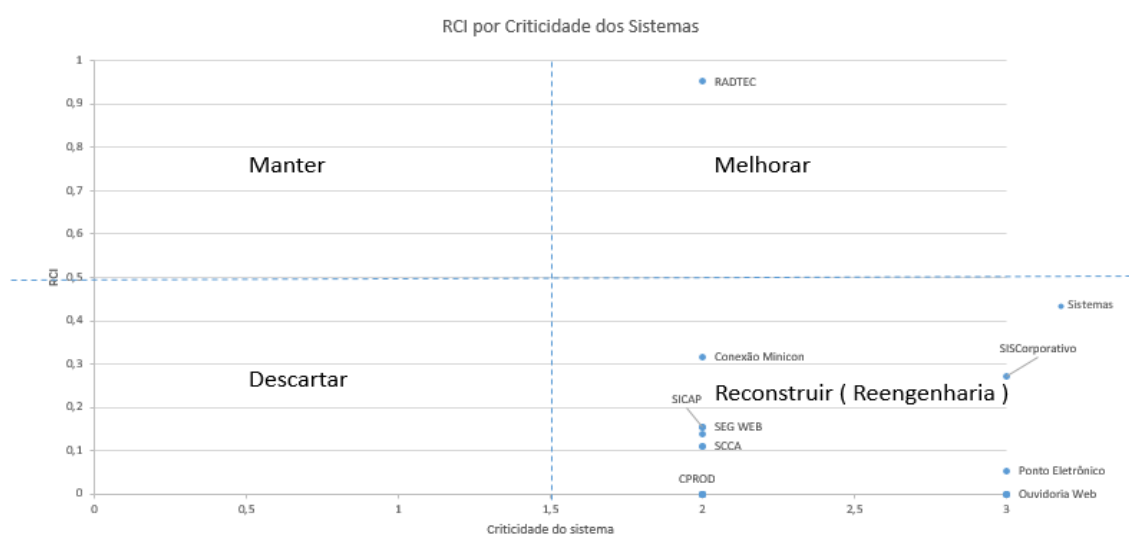
**Tabela 48:** Valores do RCI e criticidade de cada sistema. Fonte: Autor.

<b>Sistemas</b>	<b>RCI</b>	<b>Criticidade do sistema</b>
ASPAR	0,111	2
CADSEI	0	3
Conexão minicon	0,317	2
CPROD	0	2
DSCOM	0	2
OuvidoriaWeb	0	3
Ponto Eletrônico	0,054	3
RADTEC	0,952	2
REPUBL	0	3
SAF	0	2
SCCA	0,111	2
SCP	0	2
SEG Web	0,154	2
SEI	0	3
SELIN	0	2

Sistemas	RCI	Criticidade do sistema
SGD	0,138	2
SICAF	0	2
SICAP	0,155	2
Siscorporativo	0,272	3
SISGESAC	0	2
SISPA	0	2
SRH	0	3

A partir dos valores presentes na Tabela 48 foi gerado a Figura 36. Ela relaciona duas variáveis que são: RCI e criticidade dos sistemas. No eixo vertical contém os valores do RCI dos sistemas, já no eixo horizontal possui sua criticidade. Os pontos representam os sistemas.

A Figura 36 foi dividida em quatro quadrantes que representam as decisões que podem ser tomadas sobre os sistemas. A melhor decisão a ser tomada para o sistema utiliza como critério o valor do RCI e sua criticidade. A decisão é determinada através do quadrante em que o sistema se localiza.



**Figura 36:** Matriz de decisão sobre os sistemas. Fonte: (adaptado, JACOBSON; 1991).

O quadrante manter corresponde aos sistemas que possuem o RCI maior que 0,5 e menor que 1,5 de criticidade, o quadrante melhorar são sistemas com RCI maior que 0,5 e criticidade maior que 1,5, o descartar possui RCI menor que 0,5 e criticidade menor que 1,5, já o reconstruir (reengenharia) possuem RCI menor que 0,5 e criticidade maior que 1,5.

Sistemas pertencentes ao quadrante manter são sistemas com um alto RCI e uma baixa criticidade e a decisão a ser tomadas sobre eles e que a organização continue mantendo as manutenções realizadas normalmente, ou seja eles não precisam de nenhuma modificação. No quadrante melhorar são sistemas com alto RCI e alta criticidade, eles devem ser melhorados a fim de aumentar a sua qualidade interna do código. No quadrante descartar são sistemas com baixo RCI e baixa criticidade, eles devem ser descartados pela organização. Já o quadrante reconstruir (reengenharia) são sistemas com baixo RCI e alta criticidade e eles devem ser reconstruídos passando por um processo de reengenharia ou engenharia reversa.

A Tabela 49 relata os quadrantes existentes no gráficos, os valores de referência de cada quadrante e a sua interpretação.

**Tabela 49:** Interpretação dos valores do quadrantes. Fonte: Autor.

Quadrante	Valor de referência	Interpretação
Manter	RCI > 0,5 & Criticidade do sistema < 1,5	Alta qualidade e baixa criticidade.
Melhorar	RCI > 0,5 & Criticidade do sistema > 1,5	Alta qualidade e alta criticidade.
Descartar	RCI < 0,5 & Criticidade do sistema < 1,5	Baixa qualidade e baixa criticidade.
Reconstruir (Reengenharia)	RCI < 0,5 & Criticidade do sistema > 1,5	Baixa qualidade e alta criticidade.

Como vários sistemas possuem valores de RCI e criticidade iguais, os pontos que representam os sistemas na Figura 36 se sobrepõem não sendo possível visualizar todos os sistemas analisados. Para uma melhor visualização de qual decisão deve ser tomada para cada sistemas os dados foram expostos na Tabela 50, que possui os sistemas referentes a cada decisão e a justificativa dessa tomada de decisão.

**Tabela 50:** Decisão a ser tomada para cada sistema. Fonte: Autor.

<b>Decisão</b>	<b>Sistemas</b>	<b>Justificativa</b>
Manter	-	Alta qualidade e baixa criticidade.
Melhorar	RADTEC	Alta qualidade e alta criticidade.
Descartar	-	Baixa qualidade e baixa criticidade.
Reconstruir (Reengenharia)	ASPAR CADSEI Conexão Minicon CPROD DSCOM OuvidoriaWeb Ponto Eletrônico REPNBL SAF SCCA SCP SEG web SEI SELIN SGD SICAF SICAP SipaWeb Siscorporativo SisGesas SRH	Baixa qualidade e alta criticidade.

Não existe nenhum sistemas que possua um alto RCI e uma baixa criticidade e por isso nenhum sistema deve ser mantidos, ou seja, não é aconselhável continuar realizando manutenções normalmente. Também não existe sistemas com baixo RCI e baixa criticidade e por isso nenhum sistema deve ser descartados ou substituídos por outros. Somente o sistema RADTEC possui alto RCI e alta criticidade e por isso é aconselhável que melhore a qualidade interna do código. Os outros 21 sistemas analisados possuem baixo RCI e alta criticidade e por isso devem ser reconstruídos passando por processos de reengenharia e engenharia reversa.



## 7 CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho surgiu por meio da necessidade do *Ministério X* de conhecer três aspectos de seus sistemas legados, sendo eles a qualidade interna do código, a documentação e os registros de manutenção corretiva. O estudo desses aspectos é uma forma de trazer uma avaliação dos sistemas legados para apoiar a organização em seus esforços de melhoria do serviço de TI.

Antes deste trabalho, as percepções da organização sobre os aspectos dos seus legados eram baseadas em questões subjetivas e agora podem ser formadas utilizando dados reais que, muitas das vezes, vão contra as percepções que os membros da organização tinham sobre os sistemas.

Dessa forma, o objetivo deste trabalho foi diagnosticar esses três aspectos dos softwares legados e a partir desse diagnóstico obter dados que possam orientar decisões a respeito da destinação dos sistemas.

Para avaliar esses três aspectos foi utilizado o procedimento técnico de estudo de caso. Esse procedimento foi conduzido seguindo as etapas: planejamento e desenho do estudo de caso, a qual compreende a caracterização do objeto de estudo e definição do protocolo de pesquisa; coleta dos dados realizada por meio da análise de documentos e entrevistas informais; análise dos dados; e a redação deste trabalho.

Para o aspecto “qualidade interna do código”, identificou-se resultados negativos. O valor calculado pelo RCI indica que vários sistemas possuem a qualidade interna do código igual a zero. Além dos resultados do RCI, outras métricas indicaram um grande número de sistemas sem testes unitários, alta duplicação de código e muitos erros.

Em relação ao aspecto, “documentação existente nos softwares legados”, foram detectados três níveis de documentação: sistemas sem documentação; sistemas com pouca documentação; e sistemas com o conjunto mínimo de documentação estabelecido pelo órgão, que são: modelo de banco de dados, descrição de casos de uso e documento de regras de negócio.

A maioria dos sistemas analisados possui o conjunto de documentação mínima. Entretanto, as métricas coletadas demonstraram que esses documentos em muitos casos estavam incompletos.

Quanto ao terceiro aspecto estudado, “registros de manutenções corretivas”, os dados retratam que os sistemas não apresentam grande demanda por manutenção corretiva. Esse aspecto é contrário à percepção que os membros do *Ministério X* possuíam sobre seus sistemas. Dos sistemas analisados, 22 possuem menos de cinco requisições de manutenção corretiva, apenas três possuem entre cinco e dez requisições, quatro possuem entre 11 e 30 e apenas um sistema possui mais que 30. Essa quantidade de registros equivale aos cinco meses em que elas foram analisadas.

Mesmo com uma pequena demanda de manutenção corretiva, nem todas as requisições representavam erros no sistema. Algumas corresponderam a erros inexistentes. Dessa forma, é interessante que a organização reavalie os procedimentos relativos à abertura de requisições de manutenção corretiva. Para solucionar esse aspecto não é necessário nenhuma modificação no sistema, mas um treinamento dos usuários para que eles utilizem melhor os sistemas, além da conscientização sobre o custo de cada requisição de manutenção corretiva que é registrada.

Das três hipóteses analisadas neste trabalho, as hipóteses H1 (o volume de manutenção corretiva realizadas nos sistemas legados é inversamente proporcional a documentação) e H2 (os sistemas que possuem um alto número de requisições de manutenção corretiva possuem uma baixa qualidade interna do código) foram refutadas. Já a hipótese H3 (os sistemas que possuem baixa qualidade interna do código possuem um baixo nível de documentação) foi comprovada.

Em relação à hipótese 1 (H1), realizando a comparação entre os dados obtidos, é observado que não existe relação entre o número de manutenções corretivas e a existência e completude da documentação. A presença da documentação em um sistema não gerou neste caso um menor número de requisições de manutenção corretiva. Por meio dos dados utilizados neste trabalho, não é viável que a organização melhore sua documentação para que o número de manutenções corretivas do sistema diminua.

Ao analisar a hipótese 2 (H2), não foi possível identificar a relação entre a qualidade interna do código e o número de manutenções corretivas. Comparando essas duas variáveis foi observado que uma baixa qualidade interna do código não gerava um alto número de manutenções corretivas. Então mesmo que os sistemas tenham uma baixa qualidade interna, atualmente, isso não vem acarretando em prejuízos para a organização. No entanto, é possível que o baixo número de manutenções corretivas não tenha sido suficiente para avaliar essa relação.

Para a hipótese 3 (H3), observa-se que a documentação dos sistemas interfere na qualidade interna. A maioria dos sistemas que apresentaram um baixo nível de documentação também apresentaram uma baixa qualidade interna do código.

Ao aplicar a matriz de decisão, foi possível observar que a grande maioria dos sistemas deveria passar por um processo de reconstrução. Para isso, deve ser avaliado o grau de reconstrução que esses sistemas necessitam, decidindo sobre a aplicação de um processo de reengenharia ou engenharia reversa.

Para apoiar a inclusão dos sistemas no novo contrato de manutenção, foram utilizados dois aspectos das métricas coletas: a qualidade interna dos sistemas e os registros de manutenção corretiva.

As métricas relacionadas ao aspecto, “análise da quantidade de requisições de manutenção corretiva registradas”, demonstraram que os possíveis sistemas a serem incluídos no novo contrato de manutenção são: REPNBL, RADCOM, CPROD, CADSEI e OTRS. Esses sistemas possuem uma quantidade significativa de requisições de manutenções corretivas, que após serem analisadas representaram em sua maioria erros encontrados pelos usuários.

A avaliação do aspecto de qualidade interna dos sistemas indicou que a maioria dos sistemas analisados seriam candidatos a serem incluídos no novo contrato de manutenção. Pois esses sistemas possuem uma baixa qualidade interna do código, o que indica que em algum momento do seu ciclo de vida o sistema poderá apresentar defeitos que geram uma grande demanda de manutenção corretiva.

Cabe destacar que a estratégia de coleta de dados, assim como os métodos utilizados, podem servir como base para aplicação em outros sistemas legados presentes em diferentes órgãos públicos federais brasileiros e instituições privadas.

Como trabalhos futuros, sugere-se a realização da avaliação dos demais tipos de serviços da modalidade de sustentação para apoiar, juntamente com os resultados deste trabalho, a tomada de decisão sobre quais sistemas entrarão na lista de sistemas sustentados.

Sugere-se também a reconstrução de alguns sistemas e a aplicação de melhorias nos procedimentos de abertura de requisições para posteriormente realizar uma nova coleta de métricas. Com isso, seria possível comparar os dados obtidos por este trabalho e os resultados da nova coleta a fim de avaliar se, no decorrer do tempo, a organização tem melhorado os aspectos dos seus legados referentes à qualidade interna do código, à documentação e às requisições de manutenção corretiva.

## Referências Bibliográficas

BASIL, V.R.; CALDIERA, G.; ROMBACHI, H.D. Goal Question Metric Approach. Encyclopedia of Software Engineering, John Wiley & Sons, New York, v. 1, p. 528-532, 1994.

BAKOTA, T.; HEGEDUS, P.; SIKET, I.; *et al.* Quality gate Source Audit: A tool for assessing the technical quality of software. *In: 2014 Software Evolution Week - IEEE Conference on Software Maintenance, Reengineering and Reverse Engineering (CSMR-WCRE)*. [s.l.: s.n.], 2014, p. 440–445.

BAKOTA, T.; HEGEDUS, P.; LADANYI, G.; *et al.* A cost model based on software maintainability. *In: 2012 28th IEEE International Conference on Software Maintenance (ICSM)*. [s.l.: s.n.], 2012, p. 316–325.

BENNET, K.H. Legacy Systems. *IEEE Software*, v.12, n.1, p.19-23, 1995.

BENNET, K. H. Automated Support of Software Maintenance. *Information and Software Technology*, v.33, n.1, pp.74-85, 1991.

BRERETON, P. *et al.* Using a Protocol Template for Case Study Planning. EASE'08, 2008, Swinton, UK, UK. *Anais...* Swinton, UK, UK: British Computer Society, 2008. p. 41–48.

BRUSAMOLIN, Valério. Manutenibilidade de Software. *Revista Digital: Comunicações Digitais e Tópicos Relacionados*, Icesp, v. 2, p. 10–15, 2004.

CAMPBELL, G. A.; PAPAPETROU, P. P. *SonarQube in Action*. Manning, 2014.

CAMPBELL, D. T.; STANLEY, J. C. *Delineamentos experimentais e quase experimentais de pesquisa*. Editora da Universidade de São Paulo, SP, 1979.

CHELF, B.; EBERT, C. Ensuring the Integrity of Embedded Software with Static Code Analysis. *IEEE Software*, v. 26, n. 3, p. 96–99, 2009.

CHIKOFFSKY, E.J.; CROSS II, J.H. Reverse Engineering and Design Recovery: A Taxonomy. *IEEE Software*, v.7, n.1, p.13-7, 1990

COUGHLAN, P.; COUGHLAN, D. Action research for operations management. *International Journal of Operations & Production Management*, v.22, n.2, p.220-240, 2002.

CROZARA, K. H. *Monitoração da qualidade de produto nas contratações de soluções de TI da Administração Pública Federal*. [S.I.], 23 ago. 2014. Monografia (Bacharelado em Engenharia de software) —Universidade de Brasília, Brasília, 2014

DIFFERDING, C.; HOISL, B.; LOTT C. Technology Package for the Goal- Question- Metrics Paradigm. Internal Report Nr. 281/96, Department of Computer Science, University of Kaiserslautern, Germany, April 1996.

EISENHARDT, K. M. Building Theories from Case Study Research. *Academy of Management Review*, v. 14, p. 532–550, 1989.

FENTON N. E.; PFLEEGER S. L. *Software Metrics: A Rigorous and Practical Approach*. Course Technology, v2. Pp.13-14-108, 1998.

GIL, A. C. *Como elaborar projetos de pesquisa*. São Paulo: Atlas, 2008.

HUNT, B.; TURNER, B.; MCRITCHIE, K. Software Maintenance Implications on Cost and Schedule. *In: 2008 IEEE Aerospace Conference*. [s.l.: s.n.], 2008, p. 1–6.

ISO/IEC 12207, *Tecnologia de Informação – Processos de ciclo de vida de software*; Rio de Janeiro ABNT out / 1998. 35 pp. OWASP. *Static Code Analysis*, 2014.

ISO/IEC. *International Standard ISO/IEC/IEEE 14764 Software Engineering - Software Life Cycle Processes - Maintenance*. Piscataway, EUA, 2006.

ISO/IEC. *Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE*. [s.l: s.n.] 2014.

IEEE Std 1219-1998, *IEEE Standard for Software Maintenance*, 1998.

JACOBSON, I.; LINDSTRÖM, F. Re-engineering of old systems to an object-oriented architecture. *SIGPLAN Notices*, v.26, n.11, pp.340-350,nov.,1991.

LOURIDAS, P. Static code analysis. *IEEE Software*, v. 23, n. 4, p. 58–61, jul. 2006.

MARCONI, M. DE A.; LAKATOS, E. M. *Fundamentos de metodologia científica*. 5. ed. São Paulo: Atlas, 2003.

MELLO, C. H. P. et al. Pesquisa-ação na engenharia de produção: proposta de estruturação para sua condução. *Produção*, v. 22, n. 1, p. 1–13, 2012.

MILLS Everaldo E. *Software Metrics*. Relatório técnico, Software Engineering Institute, SEI - Carnegie Mellon University. pp13- 14-101, 1988.

SCHAIDT, L. P. M.; RODRIGUES Y. S.; *Avaliação de manutenibilidade de software no contexto de um órgão público federal*. Brasília 2015.

MORESI, E. *Metodologia da pesquisa*. Universidade Católica de Brasília, 2003.

OMAN, P.W.; COOK, C.R. The Book Paradigm for Improved Maintenance. IEE Software, v. 7, n.1, p39-45, 1990.

PIGOSKI, T. M. Practical Software Maintenance: Best Practices for Managing your Software Investment. Wiley Computer Publishing, 1996.

PLOSCH, R.; GRUBER, H.; HENTSCHEL, A.; *et al.* On the Relation between External Software Quality and Static Code Analysis. *In: 32nd Annual IEEE Software Engineering Workshop, 2008. SEW '08.* [s.l.: s.n.], 2008, p. 169–174.

PRADO, E. P. V.; CRISTOFOLI, F. Resultados da Terceirização da tecnologia da informação em organizações brasileiras. *Gestão & Regionalidade*, v. Volume 28, n. 84, p. PP–77–88, 2012.

PRESSMAN, R. S. Engenharia de Software. McGraw Hill Brasil, 2011.

RAMOS, C.S.; OLIVEIRA, K.M.; ANQUETIL, N. Legacy software evaluation model for outsourced maintainer. *In: Eighth European Conference on Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings.* [s.l.: s.n.], 2004, p. 48–57.

RAMOS, C.S.; OLIVEIRA, K.M.; ANQUETIL, N. Conhecendo Sistemas Legados através de Métricas de Software, 2004. CSMR 2004. Proceedings. [s.l.: s.n.], 2004, p. 1–15.

RAMOS, E.S.; BRASIL, M.M.A. Um mapeamento sistemático sobre padrões de Software para reengenharia de sistemas. P1-10, 2011.

RASHID, A.; WANG, W.Y.C.; DORNER, D. Gauging the Differences between Expectation and Systems Support: The Managerial Approach of Adaptive and Perfective Software Maintenance. *In: Fourth International Conference on Cooperation and Promotion of Information Resources in Science and Technology, 2009. COINFO '09.* [s.l.: s.n.], 2009, p. 45–50.

RUNESON, M. H. **Guidelines for Conducting and Reporting Case Study Research in Software Engineering.** Department Computer Science, Lund University, Box 118, SE-221 00 Lund, Sweden, 2008.

SANTOS, L. G. L; Um método derivado do GQM para auxiliar a gestão de sistemas de informações legados. Brasília. Julho 2008.

SIEBRA, C.A.; CAVALCANTI, A.; SILVA, F.Q.B.; *et al.* Applying Metrics to Identify and Monitor Technical Debt Items during Software Evolution. *In: 2014 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW).* [s.l.: s.n.], 2014, p. 92–95.

SNEED, H.M. A cost model for software maintenance evolution. *In: 20th IEEE International Conference on Software Maintenance, 2004. Proceedings.* [s.l.: s.n.], 2004, p. 264–273.

SNEED, H. M. Planning the Reengineering of Legacy Systems. IEEE Software, v.12, n.1, pp.24-34, 1995.

SOMMERVILLE, I. Software Engineering (International Computer Science Series). 5ª Edição. Addison-Wesley Pub Co. 1995.

SOLINGEN, R. v. Goal-Oriented Software Measurement in Practice: Introducing Software Measurement in Schlumberger Retail Petroleum Systems. Master Thesis Report, Schlumberger RPS, Netherlands, 1995.

SONAR. SonarQube Disponível em: <<http://docs.sonarqube.org/display/SONAR/Metric+definitions>>. Acesso em: 18 junho. 2015.

THIOLLENT, M. Metodologia da pesquisa-ação. 2. ed. São Paulo: Cortez : Autores Associados, 1986.

THIOLLENT, M. Metodologia da pesquisa-ação15. ed. São Paulo: Cortez, 2007.

WARDEN, R. Re-engineering - A Practical Methodology With Commercial Applications. in Applied Information Technology 12 (Software Reuse and Reverse Engineering in Practice). (P. A. V. Hall, ed.) - Chapman@Hall. 1992.

WARD, M. P.; BENNETT, K. H. Formal Methods for Legacy Systems. Journal of Software Maintenance: Research and Practice, v.7, n.3, pp.203-219, 1995.

YIN, Robert K. - Case Study Research - Design and Methods. Sage Publications Inc., USA, 1989.

YIN, R. K. Estudo de caso: planejamento e métodos. 2. ed. Porto Alegre: Bookman, 2001.