

UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Uma Arquitetura para Avaliação de Saúde de Ecossistemas de Software

Iuri Andrade Carvalho

Juiz de Fora
2018

Iuri Andrade Carvalho

Uma Arquitetura para Avaliação de Saúde de Ecossistemas de Software

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial a obtenção do grau de Mestre em Ciência da Computação.

Orientadora: Dra. Fernanda Cláudia Alves Campos.

Coorientadora: Dra. Regina Maria Maciel Braga Villela.

Juiz de Fora
2018

Ficha catalográfica elaborada através do programa de geração automática da Biblioteca Universitária da UFJF, com os dados fornecidos pelo(a) autor(a)

Andrade Carvalho, Iuri.

Uma Arquitetura para Avaliação de Saúde de Ecossistemas de Software / Iuri Andrade Carvalho. -- 2018.

113 f.

Orientadora: Fernanda Cláudia Alves Campos

Coorientadora: Regina Maria Maciel Braga Villela

Dissertação (mestrado acadêmico) - Universidade Federal de Juiz de Fora, ICE/Engenharia. Programa de Pós-Graduação em Ciência da Computação, 2018.

1. Ecossistemas de Software. 2. Qualidade. 3. Saúde de ECOS.
I. Cláudia Alves Campos, Fernanda, orient. II. Maria Maciel Braga Villela, Regina, coorient. III. Título.

Iuri Andrade Carvalho

Uma Arquitetura para Avaliação de Saúde de Ecossistemas de Software

Dissertação apresentada ao Programa de Pós-graduação em Ciência da Computação, do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora como requisito parcial a obtenção do grau de Mestre em Ciência da Computação.

Aprovada em 19 de Setembro de 2018.

BANCA EXAMINADORA

Profa. D.Sc. Fernanda Cláudia Alves Campos - Orientadora
Universidade Federal de Juiz de Fora

Profa. D.Sc. Regina Maria Maciel Braga Vilella – Coorientadora
Universidade Federal de Juiz de Fora

Prof. D.Sc. Rodrigo Pereira Santos
Universidade Federal do Estado do Rio de Janeiro

Prof. D.Sc. José Maria Nazar David
Universidade Federal de Juiz de Fora

“...não recue ante nenhum pretexto, porque o mundo tentará te dissuadir. ”
Friedrich Nietzsche.

AGRADECIMENTOS

Agradeço primeiramente a Deus, por se provar maravilhoso e presente através da ciência, onde muitos acreditam provar que Ele não existe. E, desta forma sutil, se fazer meu companheiro revitalizador durante esta jornada.

À minha namorada e companheira de vida, Beatriz, por ser meu norte magnético. E, sendo assim, nunca me permitir desviar do azimute a seguir. Por ser conforto e acalento, nos momentos bons e ruins. E, principalmente, por não desistir de mim, em momento que até mesmo eu queria desistir.

Aos meus pais, pois mesmo distantes, me forneceram apoio e incentivo.

Às minhas orientadoras, Fernanda e Regina, por terem me acompanhado neste crescimento, e por serem responsáveis por conseguir desenvolver este trabalho.

A elas e a todos os professores do mestrado, por serem fontes inesgotáveis de conhecimento e estarem sempre dispostos a compartilhá-lo comigo.

Aos membros da banca, a atenção e disposição na correção deste trabalho e poder contribuir imensamente com ele.

Aos amigos que fiz durante esta caminhada: Marcelo, Pedro, Phillipe, Márcio, Jade, Cláudio, Camila, André, Leonardo, Lenita, Heleno, Maria Luiza, Laura, Marcão, Luiz, Tati.
#TeamNenc #BandaOlar

Aos meus amigos distantes, Heitor e Karina, por entenderem minhas ausências e manterem sempre o incentivo.

À universidade Federal de Juiz de Fora, pelo apoio e fornecimento de estrutura necessária para desenvolvimento deste trabalho.

À CAPES pelo apoio financeiro através do fornecimento de bolsa de estudos.

A todos os envolvidos nessa conquista, que contribuíram direta e indiretamente, meu muito obrigado.

RESUMO

A complexidade do ambiente de Ecosistemas de Software (ECOS), onde o gerenciamento de processos e projetos não é trivial, traz grandes desafios para o controle de qualidade. Em acréscimo, os investimentos feitos pelas empresas e desenvolvedores externos podem ser perdidos caso o ECOS deixe de existir. Todos estes fatos são referentes à saúde de um ECOS, que é a capacidade na qual este se mantém e expande ao longo do tempo diante de desafios. Com base neste problema, foi proposta a arquitetura Heal Me, cujo objetivo é analisar a saúde de ECOS. No contexto desta pesquisa, foi efetuado um mapeamento sistemático, pelo qual foram identificadas as soluções atuais, assim como um conjunto de métricas para avaliação de saúde. Estas métricas foram avaliadas através de um estudo observacional, formalizadas e automatizadas como regras semânticas. Com estes recursos, foi desenvolvida uma ontologia de domínio, para analisar o ambiente de ECOS e aplicação das regras semânticas para análise de saúde. Sobre a base ontológica, foi proposta e desenvolvida a arquitetura Heal Me, uma arquitetura de software capaz de efetuar análises de saúde dos ECOS de forma automatizada. Uma vertente importante da automação propiciada é a utilização de APIs de repositórios para captura automática de alguns desses dados. Foi efetuado um estudo de caso com especialistas para avaliar a eficácia das análises deste primeiro esforço de implementação. Os resultados obtidos apontam indícios para a possibilidade da utilidade da arquitetura em seu objetivo, assim como vulnerabilidades a serem tratadas nas próximas versões.

Palavras-chave: Saúde, Qualidade, Ecosistemas de Software, ECOS.

ABSTRACT

The complexity of the Software Ecosystems (SECO) environment, where process and project management are not trivial, brings major challenges to quality control. In addition, investments made by companies and external developers may be lost if ECOS expires. All these facts refer to the health of a SECO, which is the capacity to be maintained and expanded over time in the face of challenges. Based on this problem, the Heal Me architecture was proposed, whose objective is to automatically analyze the health of a SECO. In the context of this research, a systematic mapping was performed, through which the current solutions were identified, as well as a set of metrics for health evaluation. These metrics were evaluated through an observational study, formalized and automated as semantic rules. With these resources, a domain ontology was developed to analyze the SECO environment and the application of semantic rules for health analysis. Heal Me architecture was proposed and developed over the ontological base, a software architecture capable of capturing SECO data and performing health analyzes in an automated way. An important aspect of the automation is the use of APIs repositories for automatic data capturing. A case study was carried out with specialists to analyse the effectiveness of this first implementation effort. The results obtained point to the feasibility of the proposed architecture, as well as vulnerabilities to be treated in the next versions.

Keywords: Health, Quality, Software Ecosystem, SECO.

LISTA DE FIGURAS

1	Taxonomia dos ECOSs.....	22
2	Gráfico da evolução das publicações	30
3	Modelo de avaliação de qualidade	34
4	Arquitetura SE-Advisor	35
5	Modelo de qualidade QuESo.....	36
6	Visão geral do framework OSEHO	38
7	Questões de pesquisa da RSL.....	40
8	Resultados do estudo.....	45
9	Característica Balanceamento de Esforço descrita em SWLR composta pelas métricas <i>número de desenvolvedores, número de comunidade e número horas de esforço conjunto</i>	46
10	Visão geral da ontologia <i>OntoHealth</i>	55
11	<i>OntoHealth DataProperty</i>	56
12	<i>OntoHealth ObjectProperty</i>	56
13	Property Chain isKeystone.....	57
14	Métricas modeladas em regras SWRL.....	57
15	SWRL para diversidade.	58
16	Processo de avaliação de saúde da arquitetura Heal Me.....	59
17	Visão geral da arquitetura Heal Me.....	61
18	Página Platform Registration.....	64
19	Tela de importação automática de dados.....	65
20	Alimentando tela de captura automática.....	66
21	Dado cadastrado na interface de cadastro de plataformas.....	66
22	Tela de cadastro de parâmetros.....	67
23	Health Overview.....	68
24	Individual Indicators.....	69
25	Gráfico individual da característica Heterogeneidade e suas respectivas métricas... 70	
26	Grau acadêmico dos participantes.....	76
27	Distribuição dos participantes dentre as áreas de atuação.....	76
28	Resultado do item “O panorama geral de saúde do ECOS foi facilmente compreendido na arquitetura Heal Me”.....	78

29	Resultado do item “Os níveis atingidos de saúde e de cada indicador individualmente foi facilmente visualizado na arquitetura Heal Me”.....	78
30	Resultado do item “A relação entre a saúde e seus indicadores, características e métricas foi facilmente visualizada na arquitetura Heal Me”.....	79
31	Resultado do item “A completude de cada métrica foi facilmente visualizada na arquitetura Heal Me”.	79
32	Resultado do item “A composição das características e indicadores demonstra a modelagem do domínio provida pela ontologia <i>OntoHealth</i> ”.	80
33	Resultado do item “Os gráficos visualizados na arquitetura Heal Me são de fácil compreensão ”	81
34	Resultado do item “Através da visualização gráfica é possível identificar corretamente a situação da saúde do ECOS avaliado.”.	81
35	Resultado do item “As métricas presentes na arquitetura Heal Me são minimamente suficientes para a avaliação de saúde de ECOS”.	82
36	Resultado do item “A tabela de métricas apresentada pela arquitetura Heal Me é de fácil compreensão e auxilia na utilização desta”	83
37	Resultado do item “O guia de uso apresentado torna simples a utilização da arquitetura Heal Me”.	83
38	Resultado do item “A arquitetura Heal Me é de fácil utilização e manuseio”.	84
39	Resultado do item “A arquitetura Heal Me é estável, não apresentando perda de informação”	84
40	Resultado do item “A arquitetura Heal Me possui poucas ou nenhuma falha”.	85

LISTA DE TABELAS

1	Resultados da pesquisa.....	29
2	Métricas de saúde da arquitetura Heal Me.....	48
3	Resultado das características da plataforma Wordpress	70
4	Resultado dos indicadores e da saúde da plataforma Wordpress... ..	71

LISTA DE ABREVIATURAS E SIGLAS

ECOS	Ecosystema de Software
UFJF	Universidade Federal de Juiz de Fora
LPS	Linhas de Produtos de Software
DDS	Desenvolvimento Distribuído de Software
API	<i>Application Programming Interface</i>
SO	Sistema Operacional
QM	Questão de Mapeamento
QP	Questão de Pesquisa
CI	Critério Inclusão
CE	Critério de Exclusão
PICOC	<i>Population, Intervention, Comparison, Outcome & Context</i>
MEDES	<i>International Conference on Management of Digital EcoSystems</i>
REST	<i>Representational State Transfer</i>
OSS	<i>Open Source Software</i>
OSEHO	<i>Open Source Ecosystem Health Operationalization</i>
GQM	<i>Goal Question Metrics</i>
SWRL	<i>Semantic Web Rules Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1.	INTRODUÇÃO	15
1.1.	MOTIVAÇÃO	15
1.2.	PROBLEMA	17
1.3.	QUESTÃO DE PESQUISA.....	17
1.4.	OBJETIVO.....	18
1.5.	JUSTIFICATIVA.....	18
1.6.	METODOLOGIA	19
1.7.	ORGANIZAÇÃO.....	20
2.	FUNDAMENTAÇÃO TEÓRICA	21
2.1.	ECOSSISTEMA DE SOFTWARE.....	21
2.1.1.	Saúde de Ecossistemas de Software.....	24
2.2.	MAPEAMENTO SISTEMÁTICO	27
2.2.1.	Planejamento	27
2.2.3.	Relatório de Estudo	30
2.2.4.	Resultados	31
2.3.	TRABALHOS RELACIONADOS	33
2.3.1.	Schugerl et al. (2009)	34
2.3.2.	Franco-Bedoya et al. (2014).....	36
2.3.3.	Jansen (2014).....	37
2.3.4.	Axelsson e Skoglund (2016)	38
2.3.5.	Amorim et al (2017).....	39
2.4.	CONSIDERAÇÕES FINAIS DO CAPÍTULO	41
3.	ARQUITETURA HEAL ME.....	42
3.1.	ESTUDO OBSERVACIONAL	42
3.1.1.	Definição	43
3.1.2.	Objetivos	43
3.1.3.	Planejamento	43
3.1.4.	Execução	44
3.1.5.	Resultados	44
3.1.6.	Formalização das Métricas.....	45
3.2.	ONTOHEALTH.....	54
3.3.	PROCESSO DE AVALIAÇÃO DA SAÚDE NA ARQUITETURA HEAL ME.....	58

3.4.	VISÃO GERAL DA ARQUITETURA HEAL ME.....	60
3.4.1.	Tecnologias Utilizadas	63
3.5.	EXEMPLO DE USO.....	64
3.6.	CONSIDERAÇÕES FINAIS DO CAPÍTULO	71
4.	AVALIAÇÃO DA PROPOSTA.....	72
4.1.	DEFINIÇÃO DO ESTUDO DE CASO.....	72
4.2.	CONDUÇÃO DA AVALIAÇÃO.....	74
4.2.1.	Planejamento	74
4.2.2.	Execução	75
4.2.3.	Caracterização dos Participantes	75
4.3.	RESULTADOS.....	77
4.4.	AMEAÇAS À VALIDADE.....	86
4.5.	CONSIDERAÇÕES FINAIS DO CAPÍTULO	87
5.	CONSIDERAÇÕES FINAIS	88
5.1.	CONTRIBUIÇÕES.....	89
5.2.	LIMITAÇÕES.....	90
5.3.	TRABALHOS FUTUROS.....	91
	REFERÊNCIAS.....	93
	Apêndice A - Termo de Consentimento de Participação	98
	Apêndice B - Formulário de Caracterização do Participante	100
	Apêndice C - Avaliação da Arquitetura Heal Me	101
	Apêndice D - Guia de Uso – Arquitetura Heal Me	105

1. INTRODUÇÃO

Este capítulo tem como objetivo introduzir a pesquisa sobre saúde em Ecossistemas de Software (ECOS). Este capítulo apresenta a contextualização do tema, as motivações da pesquisa, seus objetivos, a questão de pesquisa, a metodologia e a organização do trabalho.

1.1. MOTIVAÇÃO

Nos últimos trinta anos, uma das práticas de engenharia de software que ganhou grande atenção foi o reúso. A redução no custo de desenvolvimento, redução de retrabalho e ganho de tempo trazidos por esta prática colocou-a em evidência (Bosch, 2009). Modelos de desenvolvimento, paradigmas de linguagens de programação e outros princípios foram criados com a intenção de facilitar e difundir o reúso. O reúso permitiu também a criação de abordagens de desenvolvimento como as Linhas de Produtos de Software (LPS) (Gimenes e Travassos 2002) (Clements e Northrop 2002). Séries de produtos desenvolvidos a partir de uma arquitetura comum tornaram-se uma alternativa interessante para organizações desenvolvedoras pela redução do tempo e, conseqüentemente, do custo de desenvolvimento (Bosch e Bosch-Shijima, 2010).

Contudo, o cenário de desenvolvimento de software vem se alterando rapidamente (Jansen et al. 2013). Com a presença praticamente ubíqua dos softwares no cotidiano, é cada vez maior e mais urgente a demanda por soluções de software eficazes (Santos 2016). Neste cenário e com o surgimento de outras abordagens de desenvolvimento como o Desenvolvimento Distribuído de Software (DDS) (Audy e Prikladnicki 2007), apenas manter uma arquitetura central não foi o bastante para algumas empresas desenvolvedoras. Estas abriram suas arquiteturas para a colaboração de desenvolvedores externos (Jansen et al. 2009). As empresas desenvolvedoras da arquitetura passaram a ser denominadas empresas mantenedoras, e suas arquiteturas passaram a ser denominadas plataformas (Bosch 2009). Surgiu assim um novo conceito de desenvolvimento, no qual várias soluções de software, empresas e desenvolvedores aderem à uma plataforma comum. Este conceito foi denominado Ecossistema de Software (ECOS) (Jansen 2013) (Jansen et al. 2009).

Este panorama traz vantagens para as empresas mantenedoras de plataformas. O aumento da abrangência de suas plataformas, atingindo um número maior de usuários contemplados por soluções de software incorporadas a estas, deve ser destacado. Também, a

redução de custos com pesquisa e desenvolvimento, uma vez que diversas soluções de software novas são desenvolvidas por empresas externas (Bosch, 2009).

Todavia, o cenário criado pelos ECOS traz novos desafios. Dentre estes, pode-se destacar a garantia da qualidade dos produtos e da plataforma de software componentes de um ECOS. A independência dos desenvolvedores externos, assim como os próprios atributos da plataforma podem influenciar na qualidade do conjunto. Além disto, pela complexidade do cenário, a garantia de qualidade no contexto de ECOS tem suas particularidades (Santos et al. 2014) (Axelsson e Skoglund 2016).

Acelerar o processo de inovação é outra vantagem da adoção desta abordagem. Além disso, a difusão do conhecimento e a reutilização de software é facilitada dentro de um ECOS. Trocas de informações, componentes e outros artefatos são atividades comuns nessas plataformas. Assim, quanto maior a abertura oferecida pela plataforma, maior será a distribuição de conhecimento entre os atores envolvidos (Jansen, 2013). A solidez e a longevidade de um ECOS são fatores críticos. Solidez se refere ao fato de que a integridade do ECOS não é afetada por fatores como problemas relacionados ao *design*, gerenciamento, e mudanças tecnológicas. A longevidade é a capacidade do ECOS de manter-se durante um longo período (Jansen 2014). A criticidade desses fatores é devido ao envolvimento de várias empresas e os elevados esforços e investimentos financeiros feitos. Para evitar a perda desses investimentos, pode haver risco de morte do ECOS ou falhas no cumprimento das expectativas dos que adotam (Berk et al. 2010). O bom funcionamento de um ECOS, sua solidez e longevidade são chamados de "saúde" (Dhungana et al. 2010) (Manikas e Hansen 2013) (Amorim et al. 2017).

Conforme descrito, o ambiente de ECOS é complexo. Várias empresas se reúnem ao redor de uma plataforma comum de software tirando proveito da difusão de conhecimento, e contribuindo com o crescimento desta. Além disso, desenvolvedores independentes também estão envolvidos no processo, pela abertura que certas plataformas permitem. Desta forma, a criticidade dos fatores que envolvem a saúde de um ECOS é notória, devido ao envolvimento de diversas empresas, e os altos investimentos financeiros e de esforço efetuados. Para evitar a perda desses investimentos, não pode existir o risco de encerramento do ECOS, ou o não atendimento das expectativas daqueles que o adotam (Bosch 2009).

Com base em um mapeamento sistemático, foi identificada a preocupação da comunidade com a automação dos processos de análise de qualidade e saúde (Carvalho et al.

2018). Devido à complexidade do ambiente de ECOS, analisar sua estrutura e componentes de forma manual torna-se trabalhoso (Schugerl et al. 2009). Em acréscimo, a coleta de dados deste ambiente também é trabalhosa devido ao volume destes dados oriundos dos diversos componentes, da própria plataforma e de outros atores envolvidos neste (Franco-Bedoya et al. 2014).

1.2. PROBLEMA

São características intrínsecas de um ECOS a complexidade da plataforma, que traz grandes desafios para todos os envolvidos, assim como a diversidade de sua rede de relacionamentos. Por isso, o bom funcionamento e a harmonia entre estes componentes é essencial. Em acréscimo, a correta interação entre as dimensões técnica, de negócios e social são determinantes para sua solidez e longevidade. Com base nestes fatores, levando em consideração que empresas e desenvolvedores independentes alheios à plataforma se associam a esta e investem dinheiro e esforço para promoção do ECOS, da sua plataforma e de seus produtos, é importante garantir que este possua a saúde adequada. Desta forma, pode-se enunciar como problema de pesquisa deste trabalho, no contexto que os prejuízos de tempo e custo causados por Ecosistemas de Software com baixos níveis de saúde, a complexidade da análise da saúde de um ECOS. E a necessidade de ferramentas automatizadas para tal avaliação.

1.3. QUESTÃO DE PESQUISA

A partir da motivação apresentada e com base no problema descrito, pode-se enunciar a seguinte questão de pesquisa:

Uma arquitetura de software baseada em métricas e regras semânticas representadas em ontologias é capaz de auxiliar e automatizar a análise e definição de saúde de Ecosistemas de Software?

Derivadas desta questão, podem ser enunciadas duas outras questões, a saber:

QP1: A arquitetura proposta funciona adequadamente?

QP2: As avaliações de saúde efetuadas pela arquitetura proposta são eficientes?

1.4. OBJETIVO

O objetivo geral deste estudo é propor a arquitetura Heal Me, uma arquitetura de software capaz de avaliar a qualidade e a saúde de um ECOS de forma genérica e semi-automatizada, analisando dados através de métricas da literatura, regras ontológicas e técnicas de visualização para melhorar sua compreensão.

Como objetivos específicos pode-se citar:

- I. Verificação do panorama da área através da literatura corrente. Para esta etapa foi proposto efetuar um mapeamento sistemático, analisando a área de pesquisa e definindo o estado da arte e abordagens correntes;
- II. Identificação e seleção de um conjunto de métricas para avaliação da saúde de um ecossistema de software. Com base no mapeamento efetuado, foi definido um conjunto de métricas que nortearão o desenvolvimento do trabalho. Este arcabouço contou também com a avaliação da pertinência destas métricas através de um estudo observacional e da formalização das mesmas;
- III. Representação do conjunto de métricas selecionados em uma ontologia de domínio, definindo as regras semânticas;
- IV. Desenvolvimento da arquitetura de forma conceitual. Para isso, foi proposta uma estrutura de camadas para viabilizar o desenvolvimento de uma primeira versão da arquitetura;
- V. Desenvolvimento do protótipo, composto pelos passos de modelagem, desenvolvimento dos serviços para fluxo e captura automatizada de dados e desenvolvimento da interface web para captura de dados e exibição dos resultados;
- VI. Avaliação da arquitetura por especialistas.

1.5. JUSTIFICATIVA

Pela complexidade de ambientes inerentes aos ECOS, a literatura atual apresenta uma carência de trabalhos com abordagens de tratamento de saúde neste contexto. Em acréscimo, aplicações de software para automatizar este processo são escassas e muitas vezes referentes a ecossistemas de domínios específicos. Com bases nestes motivos, a proposta da arquitetura

Heal Me foi idealizada e desenvolvida para contribuir com o domínio de saúde de ECOS e buscar ferramentas para tratar os desafios encontrados nesta área.

1.6. METODOLOGIA

A metodologia deste estudo foi definida em cinco passos principais:

- I. Mapeamento sistemático;
- II. Definição e avaliação das métricas de avaliação de saúde através de um estudo observacional;
- III. Definição de uma arquitetura para a avaliação de saúde de ECOS baseada em uma ontologia;
- IV. Desenvolvimento e implementação da ontologia, da arquitetura e disponibilização da arquitetura *online*;
- V. Avaliação da arquitetura desenvolvida através de um estudo de caso.

O mapeamento sistemático foi a condução de um estudo secundário com base na literatura atual, com objetivo de localizar as abordagens e métricas mais comuns para avaliação de qualidade e saúde no contexto de ECOS.

Com o conjunto de métricas identificadas no mapeamento, foi realizado um estudo observacional para avaliação e definição de suas aderências ao contexto. Após definidas e avaliadas, as métricas foram formalizadas (ABNT 2003).

Com as métricas formalizadas, foi definida a ontologia do domínio de ECOS com propriedades para análise de saúde destes, contendo as classes de interesse que representam o ambiente. As métricas formalizadas foram definidas para serem automatizadas através de regras semânticas. A arquitetura foi desenvolvida com base no modelo multicamadas, utilizando microserviços para a transmissão de dados. As comunicações com repositórios e usuários acontecem através de *Application Programming Interface* (APIs) e de um *frontend* web desenvolvido.

Por fim, a avaliação da arquitetura ocorreu através de um estudo de caso com a análise da saúde de três ECOS efetuada por especialistas. Esta avaliação tem por objetivo detectar se a arquitetura proposta de fato é capaz de analisar a saúde de um ECOS, e apresentar os resultados de forma compreensível para os usuários.

1.7. ORGANIZAÇÃO

Este trabalho está distribuído em 5 capítulos. O capítulo 2 apresenta os pressupostos teóricos que embasaram a pesquisa, a saber: Ecossistemas de Software, suas atribuições, ambientes e relacionamentos; qualidade em ECOS e sua complexidade; saúde em ECOS, seus indicadores, características, métricas e abordagens de definição. Este capítulo traz também o mapeamento sistemático efetuado para iniciar este trabalho, assim como os trabalhos relacionados. O capítulo 3 apresenta a descrição da arquitetura Heal Me, seus componentes e a descrição de seu desenvolvimento. O capítulo 4 apresenta a avaliação da arquitetura, composta pela descrição dos estágios efetuados, a execução e os resultados obtidos. Por fim, o capítulo 5 destaca as contribuições da pesquisa efetuada, suas atribuições, limitações e trabalhos futuros. Ao final, são apresentadas as referências bibliográficas utilizadas na composição deste trabalho.

2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão apresentados os fundamentos teóricos da pesquisa. Serão descritos os ambientes de ECOS, sua utilidade, assim como os processos e a complexidade da sua avaliação de saúde. Serão descritas também as tecnologias utilizadas na pesquisa, o mapeamento sistemático que originou a pesquisa e trabalhos relacionados.

2.1. ECOSSISTEMA DE SOFTWARE

Bosch e Bosch-Sijtsema (2010) definem ECOS como:

“Uma plataforma de software, um conjunto de desenvolvedores internos e externos e uma comunidade de especialistas de domínio a serviço de uma comunidade de usuários que compõem elementos de soluções relevantes para satisfazer suas necessidades.”.

O conceito de ECOS evoluiu a partir do conceito de Linha de Produto de Software (LPS) (Gimenes e Travassos 2002) (Clements e Northrop 2002), no qual uma arquitetura de referência era utilizada por uma organização desenvolvedora. Tal arquitetura, ao ser disponibilizada para desenvolvedores externos, passou a ser denominada plataforma, e o conjunto de desenvolvimento estabelecido sobre esta foi denominado ECOS (Bosch 2009).

De forma resumida, Jansen et al. (2013) definem ECOS como sendo “no mais alto nível de abstração, uma estrutura de organizações relacionadas por meio de software ou de um conceito relacionado a softwares”. Em acréscimo, afirmam que um ECOS é parte integrante de um ecossistema de negócios, devido a todos os relacionamentos que a empresa mantenedora possui com seus fornecedores, clientes e outros sujeitos. Estes relacionamentos estão baseados ou possuem alguma dependência da plataforma e aplicações envolvidas.

Bosch (2009) definiu uma taxonomia inicial para classificar os ECOS. Esta taxonomia define três categorias de ECOS e três tipos de plataformas como mostra a Figura 1.

Programação para usuário final	MS Excel Mathematica VHDL	Yahoo! Pipes MS PopFly Google Documentos	Nada até agora
Aplicações	MS Office	SalesForce eBay Amazon Ning	Nada até agora
Sistema Operacional	MS Windows Linux Apple OS X	Google AppEngine Yahoo! developer Coghead Bungee Labs	Nokia S60 Palm Android iPhone
Categoria Plataforma	Desktop	Web	Móvel

Figura 1. Taxonomia dos ECOSs (Bosch 2009).

De acordo com a taxonomia ilustrada na Figura 1, existem três categorias de ECOS. Os ECOS centrados em Sistema Operacional (SO) são aqueles onde a plataforma é o SO de determinado dispositivo. Os ECOS centrados em aplicações são aqueles que um determinado aplicativo de software caracteriza a plataforma e recebe o desenvolvimento externo. Por fim, os ECOS construídos pelos próprios usuários finais, onde estes desenvolvem aplicações próprias e independentes, e posteriormente a disponibilizam para colaboração externa. A taxonomia ainda provê a classificação das plataformas em três tipos: desktop, onde a plataforma é executada em computadores comuns e servidores; web, onde a plataforma é um aplicativo ou portal web; mobile, onde a plataforma é executada em dispositivos mobile como smartphones, tablets, etc.

Cada plataforma é comumente mantida por uma empresa mantenedora, que é denominada organização central (Keystone). Desta forma, a arquitetura da plataforma, assim como tecnologias empregadas e outras informações de infraestrutura, são de responsabilidade da Keystone (Santos et al. 2014). Como exemplo, pode-se citar o ecossistema móvel Android, que é mantido pela empresa Google (Android 2018).

Um dos principais intuitos de um ECOS é a aderência de produtos de software à plataforma (Bosch e Bosch-Sijtsema 2010). Desenvolvedores externos à Keystone desenvolvem aplicações personalizadas, que irão atender as necessidades de diversos tipos de usuário. Este cenário é vantajoso para a Keystone por dois motivos. O primeiro deles é o aumento da abrangência da plataforma. Quanto maior o número de aplicações suportadas pela plataforma, maior é o número de usuários atingidos por esta (Bosch 2009). O segundo motivo é a redução de custos com pesquisa e desenvolvimento. Para a Keystone, o esforço de manter

a plataforma e desenvolver todos os aplicativos para atender os mais diversos tipos de usuário seria custoso. Contudo, com o desenvolvimento distribuído entre as empresas e desenvolvedores externos, este custo também é distribuído, além da maior proximidade destas como os usuários, facilitando assim o atendimento das necessidades destes (Bosch2009).

Diversos fenômenos e desafios envolvem um ECOS. Estes podem incluir os motivos que levam um ECOS a aparecer e desaparecer, a definição e monitoramento de seu escopo, regras e características. Para entender melhor a ocorrência e implicações destes fenômenos, podemos visualizar um ECOS em três dimensões (Barbosa et al. 2013). Estas dimensões são:

- Dimensão técnica: com foco na plataforma, suas tecnologias e infraestrutura, assim como seus domínios, processos e variação gerencial;
- Dimensão de negócios: com foco no fluxo de conhecimento. Este conhecimento pode ser composto por artefatos, recursos e informações para definição de negócios, inovação, e planejamento estratégico;
- Dimensão social: com foco nos *stakeholders* para definição do balanceamento entre objetivos propostos e realizados, a promoção do ECOS, e conhecimento.

A dimensão técnica tem como objetivo selecionar os alvos da plataforma, para englobar e contextualizar os desenvolvedores e disponibilizar um arcabouço de conhecimento, para suportar as atividades de desenvolvimento. Desta forma, abarca atividades de planejamento e tomada de decisão afetando propriedades do ECOS como conceitos arquiteturais e modelos de desenvolvimento.

Através da definição do contexto da plataforma, definição de regras e características, a dimensão de negócio é capaz de definir o fluxo de conhecimento através dos atores envolvidos no ECOS. Através de analogias com outros ecossistemas, esta dimensão é capaz de capturar e reaproveitar métodos e modelos de organização, classificação e evolução. Mapeando características semelhantes torna-se viável o reuso propiciado por estas analogias.

Um dos fatores envolvidos no sucesso de um ECOS é quanto a keystone está engajada na dimensão social. São exatamente os *stakeholders* os responsáveis pela evolução e definição das decisões a serem tomadas pela plataforma, equipes de desenvolvimento e dos produtos componentes do ECOS. A preocupação com tal cenário é determinante para a correta evolução da plataforma e atendimento das demandas necessárias (Bosch 2017).

Visualizar estas três dimensões do ECOS permite o melhor entendimento não só da complexidade de seu ambiente, mas também das atitudes que devem ser tomadas pela Keystone para garantir a longevidade da plataforma, sua solidez e competitividade. Solidez se refere ao fato de que a integridade do ECOS não é afetada por fatores como problemas relacionados ao *design*, gerenciamento, e mudanças tecnológicas. A longevidade é a capacidade do ECOS de manter-se durante um longo período. Competitividade refere-se a expansão da abrangência da plataforma diante de concorrência (Jansen 2014). Questões como qualidade e saúde são diretamente afetadas por fatores observados dentre as dimensões. É correto afirmar então que tais dimensões participam da responsabilidade não só do surgimento do ECOS, mas por seu ciclo de vida e a continuação de sua existência (Barbosa et al. 2013).

Todavia, as empresas externas, empresas que desenvolvem softwares aderentes à plataforma, são independentes da Keystone. Desta forma, esta não tem controle sobre modelos e processos de desenvolvimento utilizados nos softwares e aplicativos aderidos à plataforma. Em contrapartida, as empresas externas estão sujeitas à infraestrutura das plataformas. Todas estas situações impactam na qualidade dos produtos de software desenvolvidos dentro dos ECOSs (Jansen 2013). Esta questão será tratada em maiores detalhes nas seções seguintes.

Em resumo, podemos definir que um ECOS é um conglomerado de desenvolvimento que possui como ponto comum a plataforma utilizada. Este conglomerado inclui a empresa mantenedora da plataforma, seus desenvolvedores, fornecedores, colaboradores e parceiros, softwares aderentes, as comunidades externas de desenvolvimento e grupos de usuários. Todo este ambiente compõe a rede de relacionamento do ECOS e suas dimensões.

2.1.1. SAÚDE DE ECOSISTEMAS DE SOFTWARE.

Como já descrito, ECOSs são ambientes complexos, compostos por diversos atores interagindo em um ambiente de desenvolvimento distribuído de software (Santos et al. 2014), com uma plataforma onde estes atores dão suporte tecnológico ao ambiente estabelecido. Além disso, desenvolvedores independentes e comunidades de usuários surgem, em adição aos desenvolvedores existentes nas keystones. Em acréscimo, novas tecnologias emergem o tempo todo, bem como novas necessidades e requisitos de software que devem ser tratados (Manikas 2016). Todos esses fatores impactam na saúde do ECOS. De acordo com Santos et al. (2014) saúde é a capacidade de um ECOS permanecer estável e expandir, atendendo às

demandas diárias. Em outras palavras, medir a saúde de um ECOS é definir o grau em que este oferece oportunidades para as companhias e os atores que dependem dele. É importante que a rede de relações de um ECOS seja forte, e que isso não diminua ou acabe. Isso se deve ao grau de investimento que existe tanto por parte da empresa que mantém a plataforma, chamada "keystone", quanto por empresas e comunidades de desenvolvimento externas (Bosch 2009).

Desta forma, pode-se definir que um ECOS saudável é aquele que mantém seu potencial produtivo e a capacidade de atrair mais interessados, sem ser comprometido por problemas de fragmentação. Ao mesmo tempo que se mantém e se expande, adota estratégias para manter seu bom funcionamento ao longo do tempo (Amorim et al. 2017). Por estes fatores, a saúde de um ECOS é um fator decisivo.

A literatura descreve cinco indicadores principais da saúde de ECOS. Dhungana et al. (2010) discutem dois deles, nomeados: sustentabilidade e diversidade. Esses indicadores estão ligados a rede ECOS de relacionamentos, sua estabilidade e sua capacidade de envolver diferentes colaboradores. Jansen (2014) discute três outros indicadores: produtividade, robustez e criação de nicho. Estes estão diretamente ligadas à produção e à possibilidade de expansão do ECOS. Como descrição individual dos indicadores, temos:

- Sustentabilidade é a habilidade do ECOS em manter e expandir sua rede de colaboração, diante de diversidades e concorrências (Santos et al. 2014);
- Diversidade é a habilidade do ECOS em englobar diferentes colaboradores e atores, e permitir que a plataforma suporte diferentes produtos, linguagens, e tecnologias, entre outros (Dhungana et al. 2010);
- Produtividade é a habilidade do ECOS acompanhar o crescimento do mercado e entregar novidades tecnológicas, de processos e de produtos a seus atores (Santos et al. 2014);
- Robustez é a habilidade do ECOS de se manter estável, mantendo sua rede de relacionamentos e arquitetura da plataforma (Santos et al. 2014);
- Criação de nicho é a habilidade de inovação do ECOS, expandindo para sua rede de colaboração, seus produtos e atores, a capacidade de acompanhar as novidades e atualidades do mercado (Dhungana et al. 2010) (Santos et al. 2014).

Para a análise dos indicadores, Franco-Bedoya et al. (2014) propuseram dividi-los em características que são definidas por métricas. Para exemplificar, o indicador de sustentabilidade tem a característica de heterogeneidade, que pode ser definida pela métrica da diversidade geográfica, entre outros. Outra maneira de analisar os indicadores é usar diretamente um conjunto de métricas (Jansen 2014). É através do processo de análise destas métricas que a definição de características e dos indicadores de saúde pode ser realizada.

Na literatura, os indicadores de diversidade, produtividade e criação de nicho não possuem subdivisões, sendo definidos diretamente através de métricas. Contudo, os indicadores sustentabilidade e robustez possuem uma estrutura hierárquica, divididos em características como descrito anteriormente. As características nas quais o indicador sustentabilidade é dividido são (Franco-Bedoya et al. 2014):

- Heterogeneidade é o nível de diversidade entre os colaboradores, empresas e atores vinculados à rede de relacionamentos do ECOS;
- Habilidade de regeneração é a característica do ECOS em se recuperar e crescer diante de desafios;
- Balanceamento de esforço é a habilidade de distribuição uniforme das contribuições dadas dentre os grupos de desenvolvedores ou atores da comunidade;
- Expertise é o nível de envolvimento e de conhecimento obtido pelos atores envolvidos na comunidade;
- Visibilidade é a capacidade de visualização e divulgação do ECOS, da plataforma e da comunidade e deste atrair contribuintes.

Já o indicador de robustez é dividido nas seguintes características (Bosch 2017):

- Interrelação é a habilidade dos nós existentes na rede de relacionamentos do ECOS se ligarem e se comunicarem;
- Agrupamento é a capacidade de identificação e classificação dos nós da rede do ECOS dentre os componentes e projetos deste;
- Consistência financeira é a habilidade do ECOS se manter com os investimentos que são recebidos, enquanto atrai novos investidores e parceiros.

O uso de ontologias e regras de inferência pode auxiliar na análise destes indicadores e seus relacionamentos (Schugerl et al. 2009). Uma ontologia pode ser definida como uma

modelagem de informações explícitas, propriedades, conceitos, classes e seus relacionamentos. Ao contrário dos bancos de dados, as ontologias nos permitem trabalhar com conhecimento incompleto por causa de seu apoio a modelos globais, uma vez que permite, através de inferências lógicas, descobrir novas conexões entre seus elementos. Assim sua representação formal permite o uso de serviços de descoberta de conhecimento e facilita a extensibilidade. A utilização de ontologias para análise de métricas de saúde de ECOS é descrita na literatura, onde requisitos não funcionais são descobertos através da análise semântica dos dados (Schugerl et al. 2009). No entanto, o uso de regras lógicas, auxiliando na combinação de indicadores é uma solução ainda pouco explorada na literatura revisada.

2.2. MAPEAMENTO SISTEMÁTICO

Para entender melhor as implicações de avaliar a qualidade e saúde de um ECOS, seu ambiente e estrutura, foi realizado um mapeamento sistemático da literatura. Este mapeamento teve como objetivo montar um arcabouço de conhecimento para as definições e processos de análise de saúde de ECOS. Outro objetivo foi entender como a literatura atual aborda e trata esta questão, buscando estabelecer o estado da arte.

Basear as respostas das questões de pesquisa em evidências é uma abordagem importante para a eficácia dos resultados. Por essa razão, Kitchenham (2004) apresenta o paradigma de engenharia de software baseado em evidências que foi utilizado para conduzir este mapeamento.

Para realizar o mapeamento sistemático, foram definidas três fases: planejamento, condução e relatório do estudo. Para a execução de todo o processo de mapeamento, foi utilizada a ferramenta Parsifal¹ (Freitas et al. 2015).

2.2.1. PLANEJAMENTO

O primeiro passo na construção do protocolo foi a definição de questões de mapeamento e pesquisa, com o objetivo de encontrar abordagens de análise de saúde e de garantia de qualidade do ECOS e seu estado da arte. Como a saúde do ecossistema pode ser considerada uma subárea de qualidade, o mapeamento foi realizado sob o tema qualidade e, posteriormente, o estudo foi especializado para a área de saúde de ECOS.

¹Disponível em: <https://parsif.al>

Três questões de mapeamento foram apresentadas: QM1: Quais são os principais locais de publicação na área? QM2: Como os periódicos são distribuídos ao longo dos anos? QM3: Quais autores são notáveis na área? Quatro questões de pesquisa foram então propostas: QP1: Quais abordagens de garantia de qualidade são usadas para o ECOS? QP2: Quais atributos de qualidade são mais usados para avaliar o ECOS? QP3: Qual modelo é o mais utilizado na avaliação da qualidade ECOS? QP4: Qual modelo ou abordagem de qualidade é mais utilizado na avaliação de saúde do ECOS?

O *Population, Intervention, Comparison, Outcome & Context* (PICOC) (Petticrew e Roberts 2008) foi definido da seguinte forma: População: soluções que abordam a garantia da qualidade em ECOS; Intervenção: atributos e processos para a qualidade ou uso das percepções dos atores ou avaliações de saúde e prosperidade; Comparação: não foram definidas comparações; Saídas: métodos, técnicas, abordagens, modelos, soluções, metamodelos, e dimensões de qualidade; Contexto: Ecossistemas de Software.

Em seguida, os critérios de inclusão definidos foram: CI1: Publicações do ano de 2009 à frente, já que a área da ECOS ganhou maior foco a partir do trabalho publicado por Bosch (2009); CI2: Publicações com acesso gratuito, devido à necessidade de avaliar seu conteúdo; CI3: Publicações da área de Ciência da Computação, devido à sua especificidade; CI4: Publicações cujo tema principal é a garantia de qualidade ECOS. Os critérios de exclusão são a negação dos critérios de.

Na sequência, as bases de pesquisa foram selecionadas. Isto decorreu considerando publicações na área de Ciência da Computação. Outro critério foi a capacidade de usar pesquisas avançadas para aplicar a *string* de busca do mapeamento. Finalmente, todas as bases selecionadas deviam ser compatíveis com a ferramenta de suporte. As bases selecionadas foram: ACM *Digital Library* (<http://dl.acm.org/>); IEEEExplorer (<http://ieeexplore.ieee.org/>); ScienceDirect (<http://www.sciencedirect.com/>); Scopus (<https://www.scopus.com/>); e Web of Science (<http://apps.webofknowledge.com/>).

Com as bases definidas, o próximo passo foi a definição da *string* de busca. Os itens do PICOC foram usados para identificar as palavras e expressões mais relevantes. Como resultado foi gerada a seguinte *string*: (quality OR "quality assurance") AND (perception OR perceptions OR attribute OR attributes OR process OR health OR evolvability) AND ("software ecosystem" OR "software ecosystems" OR ECOS OR "software digital ecosystem" OR "software digital ecosystems"). Esta é uma *string* genérica considerando que cada base

possui sua própria sintaxe de pesquisa. Para realizar efetivamente as buscas, a *string* foi adaptada para cada base, mantendo sua estrutura básica.

Um parâmetro importante para avaliar a eficácia da busca é a definição de artigos de controle. Eles devem ser artigos importantes na área. Isto é útil para avaliar se a busca está refletindo corretamente os objetivos propostos. Cinco artigos de controle foram definidos para este mapeamento sistemático. Tais artigos também foram apresentados por (Manikas 2016) como importantes artigos de garantia de qualidade em ECOS. Os artigos selecionados são: Schugerl et al. (2009), Hmood et al. (2012), Jansen (2013), Stefanuto et al. (2011), Franco-Bedoya et al. (2014). A fase executada em seguida foi a condução do mapeamento sistemático.

2.2.2. Condução

Nesta fase, cada base foi acessada e a *string* de busca foi executada. Os resultados foram salvos em arquivos *bibtex* e importados para a ferramenta de suporte Parsifal. As buscas foram realizadas entre agosto e setembro de 2016 e atualizadas em maio de 2017. O número total de estudos encontrados foi de 109. A Tabela 1 apresenta a distribuição desses resultados por bases.

Bases	Número de Artigos
ACM	16
IEEEExplorer	12
ScienceDirect	9
Scopus	56
Web of Science	16
Total	109

Tabela 1: Resultados da pesquisa.

Entre esses artigos, 29 eram duplicados e foram automaticamente excluídos pela ferramenta Parsifal, resultando em 80 artigos. Em seguida, os títulos e resumos foram lidos e os critérios de inclusão e exclusão aplicados. Nesse processo, 57 arquivos foram excluídos, restando 23 para análise completa do texto. Após esta etapa, 11 artigos foram excluídos, resultando em 12 artigos classificados como relevantes para a área de análise de saúde e garantia de qualidade de ECOS. Refletindo a efetividade da busca, dentre os trabalhos relevantes estavam os artigos de controle, demonstrando que a *string* de busca é consistente com os objetivos do estudo.

Uma vez que os artigos tenham sido selecionados, é possível avaliar a área analisando seu conteúdo. Esta atividade faz parte da terceira etapa do mapeamento sistemático, o relatório do estudo.

2.2.3. RELATÓRIO DE ESTUDO

Analisando a distribuição das publicações podemos responder a primeira questão (QM1): Quais são os principais locais de publicação na área? Observa-se uma distribuição generalizada dos locais, uma vez que apenas dois trabalhos foram publicados na mesma conferência, os anais da *International Conference on Management of Digital EcoSystems - MEDES*. Outra nota importante é o fato de 10 trabalhos terem sido publicados em anais e apenas dois em periódicos. Essa situação denota que a área ainda não está consolidada, pois tem-se poucos trabalhos em periódicos.

Para analisar a evolução da área, podemos observar o gráfico da Figura 2, onde é apresentado o número de publicações por ano desde 2009. Com esses dados, é possível responder a segunda pergunta (QM2): Como os artigos são distribuídos ao longo dos anos? É possível visualizar a crescente importância da área nos últimos quatro anos. Este é um resultado esperado, uma vez que a abordagem de ECOS está cada vez mais presente no contexto do desenvolvimento de software atual (Manikas 2016). No entanto, entre 2011 e 2012 há uma descontinuidade no crescimento, uma vez que o número de publicações é menor em 2012 do que em 2011. Essa situação pode apontar para a imaturidade da área. Dessa forma, podemos considerar que talvez os conceitos básicos de ECOS referentes à saúde e qualidade não estejam totalmente estabelecidos.

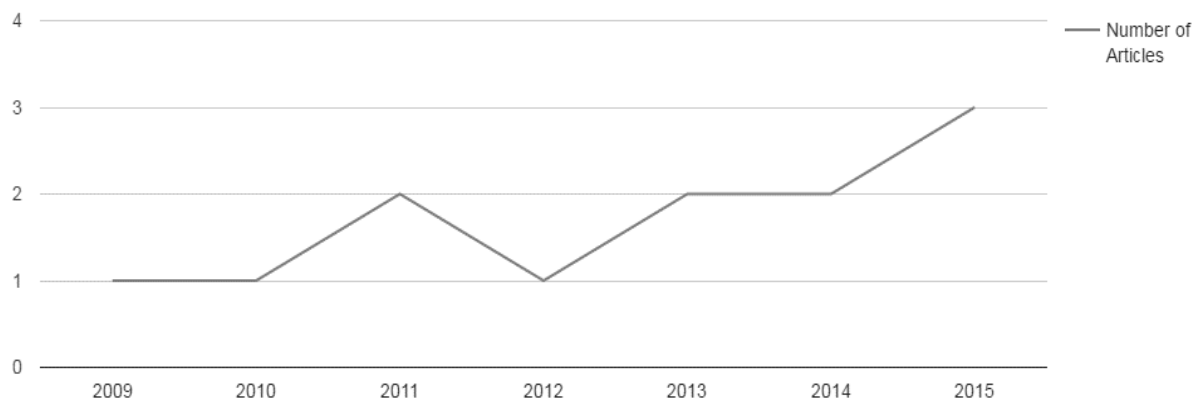


Figura 2: Gráfico da evolução das publicações.

Por fim, podemos analisar a distribuição das publicações por autor, respondendo à terceira questão (QM3): Qual autor ou autores são mais destacados na área? Ao observar os resultados, uma dispersão pode ser detectada. Entre os doze artigos selecionados, apenas os publicados por Alves e Pessôa (2010), Stefanuto et al. (2011) e Alves et al. (2015) têm autores ou coautores em comum. Os demais artigos são de diferentes autores.

Portanto, a partir desse mapeamento sistemático, há evidências de que a área ainda está emergindo e não é totalmente explorada, considerando o pequeno número de trabalhos sobre a análise de saúde e garantia de qualidade de ECOS. Além disso, identificamos a importância da destas áreas no contexto da pesquisa de ECOS. Esses resultados são detalhados na seção seguinte.

2.2.4. RESULTADOS

Para responder a primeira pergunta de pesquisa (QP1) - Quais abordagens de qualidade são usadas para o ECOS? - várias abordagens diferentes podem ser mencionadas. Entre os trabalhos estão modelos de qualidade, QuESo (Franco-Bedoya et al. 2014), CoCoADvISE (Lytra et al. 2015), SE-Advisor (Schugerl et al. 2009) e o metamodelo SE-Equam (Hmood et al. 2012). Como estruturas de qualidade, podemos citar a BISA (Kajan et al. 2011). Em Alves e Pessôa (2010) o framework denominado 5CQualiBr é apresentado, e o framework PRO2PI-MFMOD está presente em Alves et al. (2015). O Modelo de Maturidade BPS é abordado em Stefanuto et al. (2011) e Alves e Pessôa (2010). Finalmente, Mhamdia (2013) apresenta uma revisão da literatura, listando vários processos de medição aplicados à ECOS a partir de uma perspectiva de qualidade, enquanto Frantz et al. (2015) apresentam a aplicação dos processos de decisão de Markov, com o mesmo objetivo. Os resultados refletem a diversidade de áreas, considerando as abordagens identificadas. Cada proposta tem suas características exclusivas, variando de modelos de maturidade a processos de tomada de decisão. No entanto, pode-se observar o uso de técnicas comuns, como a análise semântica.

Considerando a segunda pergunta de pesquisa (QP2) - Quais atributos de qualidade são mais utilizados para avaliar o ECOS? - pode-se destacar os trabalhos de da Silva Amorim et al. (2014), a saber: comunicação, maturidade do trabalho em equipe, tecnologia e integração. Esses atributos, que estão ligados à plataforma de software do ECOS, também são abordados por Franco-Bedoya et al. (2014), Jansen (2013) e Jansen (2014). Esses três trabalhos também compartilham outros atributos, como sustentabilidade e abertura. Identificar

esses atributos é de grande importância, já que eles são a base dos principais modelos de qualidade. Desta forma, é evidente que tais atributos são importantes para uma avaliação de saúde e garantia de qualidade no contexto de ECOS.

A terceira pergunta de pesquisa (QP3) - Qual modelo é o mais utilizado na avaliação da garantia de qualidade da ECOS? - pode ser respondida a partir da análise bidimensional. A primeira dimensão é a recorrência dos modelos sem periódicos. É possível destacar o modelo de maturidade apresentada por Stefanuto et al. (2011), que é direta ou indiretamente abordado em três artigos. A segunda dimensão analisada foi o número de citações dos artigos nas bases. Como resultado, podemos apresentar Franco-Bedoya et al. (2014) e Schugerl et al. (2009) como os modelos de qualidade mais citados, e novamente Stefanuto et al. (2011) como o modelo de maturidade mais citado.

Finalmente, a quarta questão de pesquisa (QP4) - Qual abordagem ou modelo de qualidade é mais usado na avaliação de saúde de ECOS? - apresentou os modelos e métricas propostos por Franco-Bedoya et al. (2014) e Jansen (2014), onde os aspectos de saúde do ECOS são abordados diretamente. Outros modelos que tratam saúde e prosperidade de maneira menos expressiva são encontrados em Schugerl et al. (2009) e Hmood et al. (2012).

Depois de analisar os resultados, podemos destacar alguns pontos. Primeiro, as pesquisas na área seguem a tendência de criar modelos para a avaliação de saúde e qualidade em ECOS. Franco-Bedoya et al. (2014) e Jansen (2013) focam a avaliação das características e atributos do ECOS através de métricas. Outra situação recorrente é a automação de processos e o uso de análise semântica para realizar as avaliações, devido à complexidade dos processos e do ambiente a ser analisado (Schugerl et al. 2009). No entanto, os artigos não abordam o ECOS de maneira genérica, em geral, apenas contextos específicos são analisados.

Além disso, a automação de processos ainda é imatura e acredita-se que ela pode ser substancialmente aprimorada através do uso de ferramentas específicas e análises inteligentes. Outra desvantagem é a apresentação dos resultados das avaliações dos modelos propostos, uma vez que existe carência de análises mais elaboradas com recursos de visualização adequados.

Como ameaça à validade do mapeamento sistemático, podemos citar as limitações apresentadas pela ferramenta de apoio (Parsifal). Como funciona com arquivos *bibtex*, as bases de dados que não suportam esse formato, como o Springer Link, foram excluídas. Esta

escolha pode reduzir o número de artigos válidos. No entanto, esta ameaça pode ser atenuada pela presença da base de busca SCOPUS, pois esta indexa artigos de outras bases. Outra ameaça é a criação e adaptação da *string* de busca. Como as palavras e frases são derivadas do PICOC, a construção correta é vital para a eficácia da pesquisa. Adaptar a *string* a cada base de dados pode alterar o foco da pesquisa, se não for feito com precisão. Para atenuar essa ameaça, especialistas foram solicitados a avaliar a *string* de busca para validá-la e melhorar sua eficácia. Outra ameaça a ser destacada é ausência de *snowballing* durante o mapeamento. Esta prática não é obrigatória, porém pode contribuir para uma melhor precisão dos resultados. É proposta a execução desta técnica como trabalho futuro, tendo em vista o objetivo original deste mapeamento.

A última ameaça que deve ser destacada é o número de pesquisadores que participaram da elaboração do mapeamento. Este mapeamento foi feito por apenas um pesquisador. Tal situação aumenta o risco de vieses presentes no trabalho. Para atenuar essa ameaça, o pesquisador foi acompanhado por especialistas e orientadores, reduzindo o risco de interpretações pessoais presentes no mapeamento.

2.3. TRABALHOS RELACIONADOS

A partir do mapeamento sistemático realizado, foi possível identificar que existem poucos estudos que tratam da saúde de ECOS nas bases pesquisadas. Encontramos 12 artigos que direta ou indiretamente abordam a saúde dos ECOSs. Dentre estes, três artigos foram destaques, nos quais indicadores de saúde e qualidade, suas métricas e métodos de avaliação são descritos. Em acréscimo foram inseridos dois trabalhos secundários correlatos ao tema. O primeiro, divulgado em importante periódico da área, descreve um mapeamento sistemático focado em garantia de qualidade em ECOS, e propõe uma agenda de pesquisa para tratamento desafios encontrados durante a pesquisa. O segundo, publicado em evento referência da área, apresenta uma Revisão Sistemática da Literatura (RSL) focada na definição de saúde, apresentando as métricas utilizadas atualmente, abordagens e métodos de avaliação e definição de saúde presentes na literatura.

2.3.1. SCHUGERL ET AL. (2009)

Este artigo apresenta uma arquitetura para análise semântica de requisitos não funcionais de software no contexto de ECOS. Ao descrever requisitos não funcionais de software como atributos de qualidade, este trabalho indica a garantia da qualidade de produtos de software através da análise destes requisitos.

Além da garantia de qualidade, o trabalho citado ainda busca avaliar a evolução do produto de software através da análise semântica das atividades de aperfeiçoamento de seus artefatos. Para isto, primeiramente o domínio dos requisitos não funcionais de software foi modelado semanticamente utilizando ontologias. A modelagem de processos de software também foi contemplada, assim como os artefatos de software. Através disto, a análise, acesso e avaliação dos atributos não funcionais de software pôde ser feita.

Para efetuar a análise, a arquitetura proposta acessa os repositórios onde estão os artefatos a serem analisados. Os artefatos como códigos-fonte, controle de versionamento e relatórios de erros são processados. Assim, é feita a análise semântica dos atributos de qualidade no nível de artefatos. Para determinar a evolutibilidade, é efetuada a rastreabilidade de itens como, por exemplo, a persistência de determinado erro. Os resultados são capazes de definir a capacidade de evolução do produto de software. A Figura 3 demonstra este processo de avaliação.

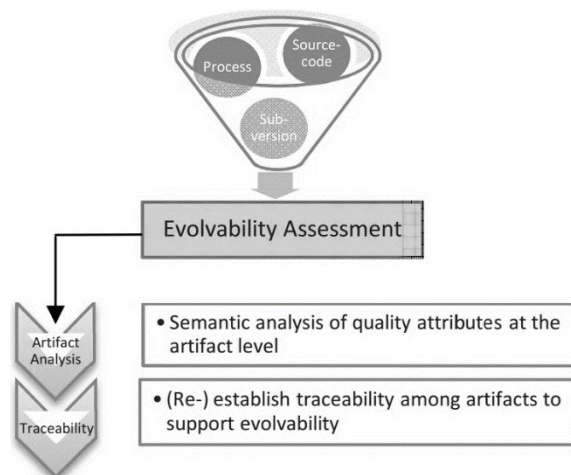


Figura 3: Modelo de avaliação de qualidade (Schugerl et al. 2009).

Contudo, uma das dificuldades de efetuar a avaliação descrita é o cenário distribuído de desenvolvimento, onde os artefatos de um mesmo produto podem estar espalhados por

diversos repositórios. Para isto, foram necessárias adequações ao modelo semântico, para que este contemplasse tal situação

As ontologias geradas para os processos descritos foram incorporadas a uma arquitetura denominada SE-Advisor. Esta utiliza serviços web do tipo *Representational State Transfer* (REST) para acessar os repositórios e efetuar as análises. Os resultados são persistidos na base de conhecimento e podem ser utilizados para monitoramento dos atributos de qualidade dentro do ECOS. A Figura 4 ilustra a arquitetura SE-Advisor.

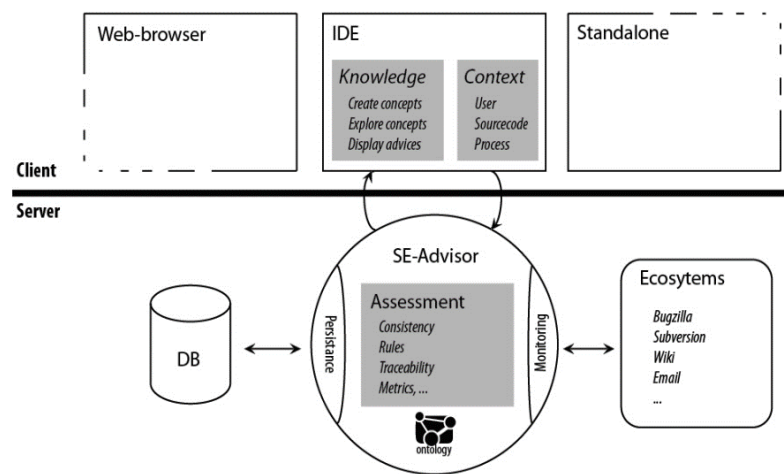


Figura 4: Arquitetura SE-Advisor (Schugerl et al. 2009).

As contribuições apresentadas no trabalho são duas. A primeira é um modelo semântico rico e unificado para obtenção de conhecimento sobre vários recursos e componentes. A segunda é a arquitetura proposta, que é capaz de prover a análise automatizada ou semi-automatizada de atributos não funcionais de artefatos de software, por meio da qual é possível efetuar a avaliação da qualidade de evolutibilidade do produto de software e, conseqüentemente, do ECOS.

Contudo, o trabalho apresentado foca especificamente na qualidade dos produtos de software, desprezando seu relacionamento com a plataforma e o restante da estrutura do ECOS. Apesar do grande poder de decisão provido pela análise semântica, na qual é utilizada uma ontologia como tecnologia base, os produtos de software são um recorte dentre toda estrutura do ECOS, que interferem diretamente nos fatores e indicadores apresentados. O trabalho citado apresenta também a utilização de análise semântica para automatizar o processo de análise de qualidade, mas não aborda a aplicação de técnicas semelhantes no processo de análise de saúde de ECOS. Esta lacuna é tratada na proposta desta dissertação.

2.3.2. FRANCO-BEDOYA ET AL. (2014)

Um ECOS pode ser visto da perspectiva de uma comunidade de atores, organizações e companhias, ou pelo foco técnico e dos aspectos sociais, definindo-o como um grupo de projetos de software e suas comunidades. A partir destas perspectivas, Franco-Bedoya et al. (2014) propõem acessar a qualidade de ECOS em seu sentido mais amplo. A partir de um ECOS denominado *Open Source Software* (OSS), a iniciativa deste trabalho buscou definir características de qualidade que são inerentes ao ECOS.

Para elencar estas características, Franco-Bedoya et al. (2014) utilizaram uma Revisão Sistemática da Literatura (RSL). Com as características definidas, foi construído o modelo, denominado QuESo. Este modelo foi composto de dois tipos de elementos: características de qualidade e medições. As características de qualidade correspondem aos atributos de software relevantes para avaliação. Estas são organizadas através de uma hierarquia, dividida em dimensões. As dimensões são a plataforma do ECOS, a comunidade componente deste e sua rede de atores. As dimensões são divididas em características e estas, por sua vez, são divididas em subcaracterísticas. Algumas das características elencadas foram extraídas de um modelo de qualidade para softwares *Open Source*, denominado QualOSS (Soto e Ciolkowski 2009). A Figura 5 ilustra o modelo QuESo.

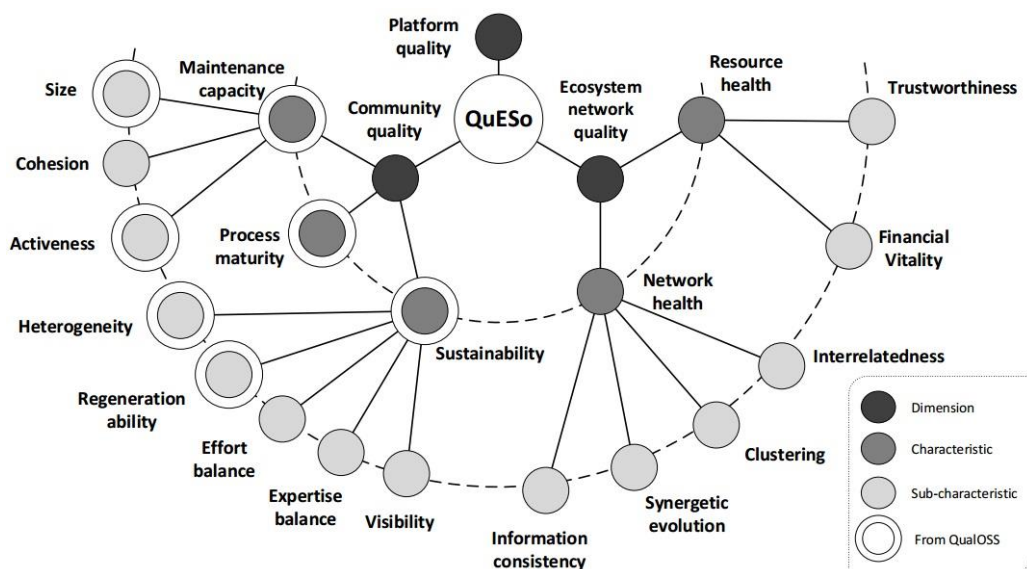


Figura 5: Modelo de qualidade QuESo (Franco-Bedoya et al. 2014).

Para exemplificar as medidas componentes do modelo pode-se citar a subcaracterística tamanho, integrante da característica capacidade de manutenção, da dimensão qualidade da

comunidade. Tal subcaracterística pode ser mensurada através do número de contribuintes do ECOS, número de membros, número de erros corrigidos dentre outros.

Este modelo mostra-se capaz de analisar a qualidade da plataforma dos ECOSs. Contudo, este não aborda a influência da qualidade da plataforma na qualidade dos produtos presentes no ECOS. Em acréscimo, o modelo e as métricas apresentadas, assim como seus indicadores e características, são apresentados de forma conceitual, não possuindo nenhuma forma de automatização.

2.3.3. JANSEN (2014)

O trabalho apresentado por Jansen (2014) tem como objetivo analisar e definir o funcionamento de ECOS *Open Source* e sua saúde, com base em três indicadores: produtividade, robustez e criação de nicho. O autor sugere que a saúde de um ECOS pode ser definida por dois fatores: longevidade e capacidade de crescimento. Outra observação descrita pelo autor é a diferença entre a saúde do ECOS e a saúde dos projetos de software. A saúde de projetos de software, tratada pelo autor como integridade, é definida por meio de métricas como: consistência de *commits*, rastreamento e correção de erros, número de releases e número de downloads. Segundo o autor, definição de saúde do ECOS depende de fatores como: rede de relacionamento e capacidade de expansão. O autor apresenta estes fatores como complexos de serem medidos.

A abordagem do autor fornece uma visão geral das métricas necessárias para definir a integridade de um ECOS. Como resultado, uma estrutura chamada *Open Source Ecosystem Health Operationalization* (OSEHO) foi gerada. A Figura 6 apresenta as métricas propostas pelo referido trabalho para avaliar a saúde utilizando três indicadores: produtividade, robustez e criação de nicho. As métricas são instanciadas no contexto teórico, e especificadas nos níveis de rede e projetos dos ECOS.

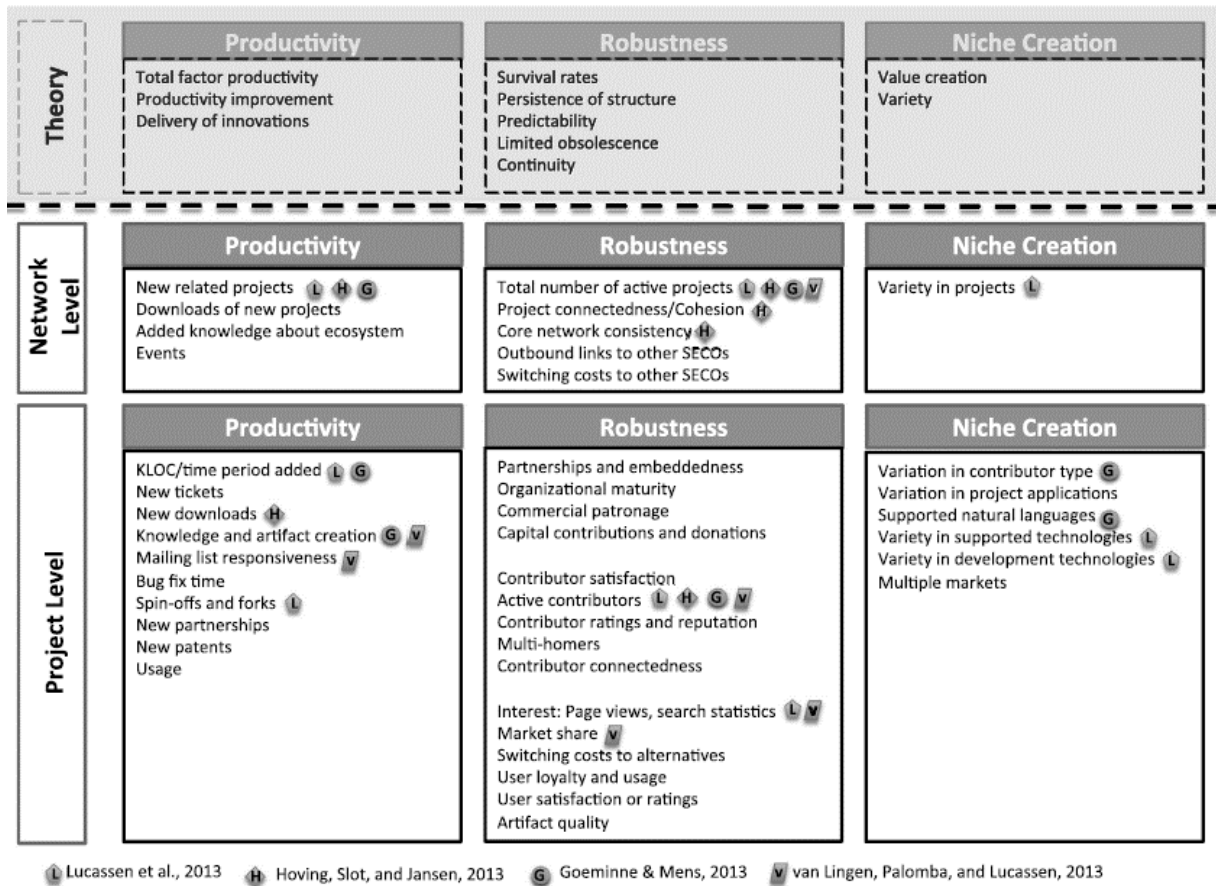


Figura 6: Visão geral do framework OSEHO (Jansen 2014).

Este trabalho apresenta métricas e processos para a definição de indicadores de produtividade, robustez e criação de nicho. No entanto, os indicadores de sustentabilidade e diversidade, importantes para a saúde de um SECO, não são abordados. Além disso, a estrutura da OSEHO é conceitual. Portanto, este trabalho não inclui a automação das métricas.

2.3.4. AXELSSON E SKOGLUND (2016)

Uma das principais características dos ECOS é a existência de diferentes atores que cooperam em torno de uma plataforma compartilhada. No entanto, não está claro quais são as implicações na qualidade do software ao passar de uma abordagem tradicional para um ecossistema, e isso está se tornando cada vez mais importante à medida que os ecossistemas emergem em domínios críticos, como aplicativos embarcados. O trabalho de Axelsson e Skoglund (2016) investigam os desafios relacionados à garantia de qualidade em ecossistemas de software e identifica quais abordagens têm sido propostas na literatura. O método de pesquisa utilizado é um mapeamento sistemático.

Após os procedimentos de planejamento e execução das buscas, o mapeamento resultou em seis artigos. As descobertas da literatura são complementadas com uma abordagem construtiva onde são identificadas áreas que merecem mais pesquisas, resultando em um conjunto de tópicos de pesquisa que formam uma agenda de pesquisa para garantia de qualidade em ecossistemas de software. Pode-se destacar nos resultados abordagens nas áreas de arquitetura de sistemas, políticas de governança para verificação e validação, propagação de conhecimento, métodos de teste e métodos gerais de qualidade. No relatório, são relacionados cada um destes conceitos com as dimensões de ECOS.

Como desafios para garantia da qualidade são listados fatos relacionados à comunicação entre fronteiras organizacionais, incentivos para o compartilhamento de informações e motivos para não o fazer e a integração final do sistema, realizada pelo usuário selecionando uma combinação específica de componentes do ambiente, o que significa que os participantes da keystone e as empresas desenvolvedoras não podem executar atividades tradicionais de garantia de qualidade, como testes de integração. O contexto de execução é até certo ponto desconhecido, e tanto a plataforma quanto as extensões devem estar preparadas para coexistir.

Apesar de abordar a saúde de ECOS, o trabalho de Axelsson e Skoglund (2016) tem foco específico em garantia da qualidade neste contexto. Em acréscimo, esta garantia é especificada para abordagens de testes e verificações, conforme apresenta a própria string de busca. Como comparativo, o trabalho desta dissertação se difere pelo foco em saúde de ECOS como especialização da garantia de qualidade. Além disso, o escopo desta pesquisa busca modelos, abordagens e métodos de forma genérica, sem especialidade em verificações e testes.

2.3.5. AMORIM ET AL (2017)

Devido à grande adoção das empresas na utilização de ECOS, a saúde torna-se um fator de extrema importância. Com base neste cenário, a pesquisa sobre saúde de ECOS tem ganhado crescente atenção. Contudo, ainda não existem pesquisas sólidas, com técnicas e abordagens consolidadas para o tratamento de saúde neste cenário. Com base nestas afirmações, Amorim et al. (2017) propõem uma RSL para identificar os principais trabalhos e as principais tendências no tratamento de saúde no contexto de ECOS. Através da Figura 7 pode-se

observar os objetivos do trabalho apresentado, através de suas questões de pesquisa e motivações.

Nr.	Research question	Motivation
RQ1	What definitions of the term “health of software ecosystem” exist?	There are several definitions for the health of software ecosystems. We aim to find out which concepts each study is using in its design.
RQ2	What are the approaches to evaluating the health of software ecosystems?	With this question, we intend to identify the approaches, methods, models or frameworks which are used for evaluating the health of software ecosystems.
RQ3	What key areas are emphasized by existing approaches as having an influence on evaluating the health of software ecosystems?	By a key area, we mean a characteristic that is common across a set of studies or processes that has an influence on the health of software ecosystems.
RQ4	Which business, technical and social practices are used by existing approaches to evaluate the health of software ecosystems?	Through this question, we hope to understand which practices have a substantial role in the software ecosystem health that is not addressed by the existing approaches of evaluating the health nowadays.
RQ5	What are the metrics used to measure the health of software ecosystems?	In this question, we intend to identify metrics commonly used in evaluation of the health of software ecosystems.
RQ6	Are there tools that support the evaluation process of the health of software ecosystems?	There are several tools that support different models or approaches of evaluation. In this sense, we would like to investigate which tools are used to support the evaluation processes of the health of software ecosystems.

Figura 7: Questões de pesquisa da RSL (Amorim et al. 2017).

Após definição dos objetivos, construção da *string* de busca e execução nas bases propostas, foram encontrados 2280 estudos relevantes, dos quais restaram 23 depois dos processos de filtragem. Dentre estes trabalhos foram coletados os dados para responder as questões propostas.

Com base nos trabalhos resultantes das buscas, Amorim et al. (2017) enunciaram uma importante definição para saúde de ECOS sintetizada de 6 definições presentes em outros trabalhos, a saber, “*Um ecossistema de software saudável tem a capacidade de manter sua produtividade e atratividade, enfrentando problemas, interrupções e junções. Ao mesmo tempo, este ECOS também monitora e implementa os avanços de suas estratégias para alcançar o sucesso ao longo do tempo.*”

Dentre os 23 trabalhos foram localizadas as abordagens de saúde propostas. Estas são caracterizadas por análise de governança, controle de projeto de desenvolvimento, criação e

monitoramento do ECOS, indicadores de performance e métricas de saúde. Dentre estas abordagens, foram extraídas 251 métricas para avaliação de saúde e ECOS. Outras técnicas de avaliação de saúde encontradas estão entre áreas chave, que aplicam métodos de pesquisa empírica, e ferramentas de avaliação.

O trabalho apresentado por Amorim et al. (2017) é uma rica fonte de informações sobre avaliação de saúde no contexto de ECOS destacando-se como um estudo secundário. A proposta apresentada nesta pesquisa visa apresentar uma arquitetura, não apenas o conjunto de métricas, para avaliação de saúde de ECOS, tratando este conceito vinculado à qualidade. Em acréscimo, é proposta uma ferramenta web, para automatizar o trabalho dos avaliadores de saúde de um ECOS.

2.4. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou o referencial teórico necessário para se compreender a extensão da pesquisa efetuada. Foram abordados os conceitos de ECOS e a complexidade de seu ambiente. Foram descritos também fatores que influenciam na sua saúde e qualidade, e como estas grandezas podem ser estruturadas. Em acréscimo, foi apresentado um mapeamento sistemático efetuado com foco na área, para descrever os principais trabalhos publicados, seu conteúdo e métodos de abordagem de saúde e qualidade buscando definir o estado da arte.

Os estudos citados na última seção apresentam modelos e algumas métricas, que podem ser o estado da arte para a definição de saúde em ECOS (Jansen 2014) (Franco-Bedoya 2014). Esses trabalhos também apresentam a importância da definição de saúde e o impacto que esse cenário tem no ambiente, rede de relacionamentos e componentes do ECOS. No entanto, nenhum deles aborda todos os aspectos da saúde citados na literatura. Estes não apresentam também uma proposta de automação do processo de avaliação através de uma arquitetura. O próximo capítulo descreverá a arquitetura Heal Me, o modelo semântico descrito através de uma ontologia, o uso e aplicação de métricas e automação do processo de avaliação da qualidade e da saúde de ECOS.

3. ARQUITETURA HEAL ME

Este capítulo apresenta a arquitetura Heal Me e seus principais componentes, dentre estes as métricas definidas para analisar a saúde de um ECOS. Além disso, são também detalhados o estudo observacional utilizado para a seleção das métricas de saúde de um ECOS a serem utilizadas, a ontologia especificada a partir da definição destas métricas. São também descritas as camadas da arquitetura, suas funcionalidades e as tecnologias utilizadas. Por fim, o processo de análise da saúde é apresentado.

3.1. ESTUDO OBSERVACIONAL

Considerando os resultados apresentados no mapeamento sistemático, a arquitetura (Carvalho et al. 2017) foi realizado um estudo observacional com o objetivo de definir as métricas mais adequadas para a saúde de um ECOS. De acordo com Yin (2015), um estudo observacional deve ser realizado quando há necessidade de informações adicionais sobre o objeto de estudo. Desta forma pode-se estimar com maior precisão os limites deste objeto que, no caso deste trabalho, tratam-se das métricas. O contexto da avaliação de saúde também pode ser melhor compreendido e enriquecido através deste estudo. Yin (2015) ainda afirma que para uma confiabilidade maior das evidências a serem observadas é aconselhado que mais de um observador seja envolvido.

O estudo foi realizado em duas etapas. Inicialmente, os dados foram organizados em forma de uma lista de métricas, para facilitar a captura da percepção dos especialistas sobre cada uma delas. Em seguida, um questionário estruturado com todas as 58 métricas identificadas na literatura após o mapeamento sistemático foi enviado para os especialistas. Cada um deles analisou individualmente as métricas. Durante o segundo passo, as respostas dos especialistas foram quantificadas atribuindo pesos a estas. Então, foi possível formalizar as métricas e definir seus componentes utilizando as informações obtidas nos resultados do estudo.

O estudo foi organizado considerando cinco fases: Definição, Objetivos, Planejamento, Execução e Resultados. A definição do estudo teve como objetivo delinear os passos iniciais da pesquisa. A fase de objetivos define os principais alvos do estudo. A fase de planejamento teve como objetivo definir a aplicação do próprio estudo, bem como os itens necessários para sua execução e as fontes de dados selecionadas. A fase de execução detalhou

as etapas para alcançar os resultados e, finalmente, na fase de resultados, a análise dos dados foi detalhada, enfatizando como os objetivos foram alcançados (Perry et al 1998).

3.1.1. DEFINIÇÃO

O escopo do estudo observacional foi definido com base no método *Goal Question Metrics* (GQM) (Basili et al. 1994): “**validar** um grupo de 58 métricas de saúde de ECOS **com o objetivo de** verificar a utilidade **com respeito a** importância de cada métrica **do ponto de vista de** especialistas **no contexto de** um ECOS”.

Para definir o escopo, a seguinte questão de pesquisa foi proposta: Quais métricas são essenciais para a avaliação da saúde de um ECOS e devem constar na arquitetura Heal Me?

3.1.2. OBJETIVOS

O objetivo do estudo observacional foi verificar a utilidade e importância de cada métrica capturada através do mapeamento sistemático do ponto de vista de especialistas no contexto do ECOS. Com base nesta verificação, as métricas mais bem avaliadas serão integradas à proposta Heal Me. Para atingir esse objetivo, o estudo foi aplicado aos sujeitos utilizando um questionário como principal instrumento. Cada sujeito avaliou as métricas apresentadas dentro da escala que permitia a priorização da importância das mesmas para a saúde de um ECOS.

3.1.3. PLANEJAMENTO

Para avaliar as métricas propostas, o estudo foi realizado considerando a experiência dos especialistas no contexto de ECOS. Participaram da avaliação três especialistas com grau de mestrado. Possuem de cinco a oito anos de experiência acadêmica, onde a experiência destes no contexto de ECOS varia de dois a cinco anos. Os sujeitos eram especialistas em engenharia de software, dois da indústria e um da academia.

A coleta de dados foi baseada em um questionário estruturado *on-line*. Cada sujeito avaliou cada métrica, considerando sua utilidade para avaliação da saúde de um ECOS dentro da seguinte escala: Essencial, Desejável e Dispensável. O uso de uma escala de múltipla escolha para responder ao questionário reflete a precisão necessária na avaliação da métrica.

3.1.4. EXECUÇÃO

A aplicação do questionário foi feita de forma online, sem interferência do pesquisador durante o processo. Todos os sujeitos responderam o questionário e não relataram dúvidas quanto ao seu preenchimento.

Os sujeitos acessaram o questionário e selecionaram individualmente as respostas, em diferentes dias e horários. Cada sujeito levou, em média, cerca de 20 minutos para completar o questionário. Um ponto importante a ser enfatizado é que todos os sujeitos têm um bom conhecimento sobre o ECOS, já tendo participado de projetos neste contexto. Após o estudo ter sido realizado, as respostas foram coletadas, analisadas e os resultados são relatados a seguir.

3.1.5. RESULTADOS

Os resultados foram analisados para avaliar a importância de cada métrica ao processo de avaliação de saúde proposto pela arquitetura Heal Me. Cada resposta foi distribuída considerando três escalas: Essencial, Desejável e Dispensável. Para realizar a análise, cada resposta recebeu um peso, respectivamente, 2, 1 e 0. Considerando que: Essencial é uma métrica obrigatória; Desejável é uma métrica que pode ser considerada; e Dispensável são as que devem ser descartadas. Os pesos demonstram a relevância de cada resposta definida de maneira numérica. Esses pesos foram totalizados em cada métrica considerando a resposta de cada sujeito. As médias foram calculadas e classificadas na seguinte escala: Essencial (média maior que 1,66); Parcialmente Essencial (média menor que 1,66 e maior que 1); Desejável (média igual a 1); Parcialmente Dispensável (média menor que 1 e acima zero); e Dispensável (média igual para zero). Os resultados são apresentados na Figura 8.

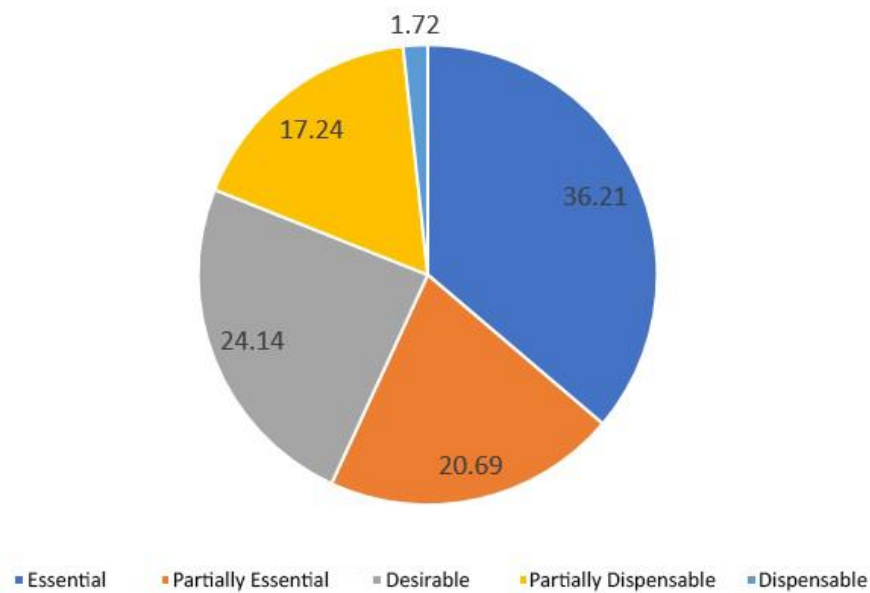


Figura 8: Resultados do estudo.

Pode-se perceber que 81,04% das métricas foram avaliadas entre Essencial e Desejável, totalizando 47 métricas. Este resultado mostra evidências de que as métricas são úteis e aderentes ao contexto proposto. Por outro lado, detectou-se que 18,96% das métricas foram avaliadas entre Parcialmente Dispensável e Dispensável, totalizando 11 métricas. Isso demonstra que novos estudos devem ser realizados para validar as evidências coletadas nesta primeira avaliação. No entanto, com esses resultados, a questão de pesquisa pôde ser respondida, considerando as 47 métricas positivas avaliadas. Estas métricas avaliadas positivamente foram mantidas na arquitetura e descritas na Tabela 2 (Carvalho et al. 2017), e as 11 métricas avaliadas negativamente foram descartadas.

3.1.6. FORMALIZAÇÃO DAS MÉTRICAS

Ao considerar as métricas extraídas do mapeamento sistemático, e que foram analisadas usando o estudo observacional, a Tabela 2 (Carvalho et al. 2017) apresenta as métricas resultantes do estudo. Cada métrica foi detalhada e seu indicador de saúde foi apontado, assim como a dimensão de ECOS à qual a métrica pertence, seu nome, sua medida e fórmula, ator a quem a métrica se refere, além de sua referência bibliográfica. Após o detalhamento das métricas, estas foram formalizadas em regras ontológicas, conforme será apresentado na seção 3.2. Um exemplo de tais regras pode ser visto na Figura 9. Vale destacar

que a forma original das métricas na literatura exhibe apenas nome e descrições das métricas, sem detalhes e formalização.

```

hasActiveDeveloper(?x, ?y) ^
hasContributedWithHours(?x, ?y) ^
seco:Platform(?x) ^
hasContributedWithCommits(?x, ?y) ^
hasGreatWorkTogether(?x, ?k) ^
hasContributedWithReleases(?x, ?y) ->
EffortBalance(?x)

```

Figura 9: Característica Balanceamento de Esforço descrita em SWLR composta pelas métricas *número de desenvolvedores*, *número de comunidade* e *numero horas de esforço conjunto*.

Para facilitar a interpretação, as métricas foram agrupadas nas três dimensões de ECOS, a saber: Técnico, de Negócios e Social (Barbosa et al. 2013). Os componentes das métricas formalizadas e os procedimentos de avaliação são: Nome, que identifica a métrica; Descrição, que mostra a utilidade da métrica; Indicador a qual a métrica pertence; Dimensão de ECOS ao qual a métrica está vinculada; Medida e Fórmula, que detalha a aplicação da métrica usando seus dados; Interpretação, que mostra o significado do resultado da métrica; Unidade para quantificar a medida do resultado; Ator, o indivíduo relacionado com a métrica; e Referência Bibliográfica de onde a métrica foi extraída. Com esses componentes, a precisão da avaliação das métricas é aumentada e os resultados refletem a realidade do contexto avaliado.

Para facilitar a obtenção dos dados a serem analisados pelas métricas, estas possuem atores vinculados conforme descrito. Estes são capazes de localizar e capturar os dados, ou estão envolvidos no ambiente no qual o dado é extraído. Os atores são:

- Gerente de Projetos: responsável pelos projetos existentes na plataforma e suas implicações;
- Gerente de Rede: responsável pelas conexões dentro do ECOS, gerenciando a conexão dos nós;
- Gerente de Comunicações: responsável pela transmissão de mensagens e outras informações dentro do ambiente do ECOS e seus canais de comunicação;

- Gerente Financeiro: responsável pelo gerenciamento de valores, patrocinadores e outros componentes referente aos custos, lucros e viabilidade financeira do ECOS;
- Colaborador: é o ator que de alguma forma contribui para a plataforma e seus projetos. Podem ser desenvolvedores, colaboradores financeiros, parceiros comerciais, etc.;
- Usuário: é o cliente final da plataforma e suas aplicações de software.

A partir da definição detalhada das métricas, uma ontologia foi utilizada para representar o conhecimento expresso pelas métricas e as características de saúde identificadas nos ECOS a serem analisados. Desta forma, a partir dos dados específicos de ECOS, é possível a descoberta de conhecimento novo a partir de relações entre os elementos destes encontrados através da execução de uma máquina de inferência. O uso de ontologias como tecnologia de descoberta de conhecimento foi escolhida pelo conhecimento prévio e habilidade do pesquisador com esta técnica. O desenvolvimento é detalhado na seção seguinte.

Número	Dimensão	Indicador	Métrica	Descrição	Medida e Fórmula	Interpretação	Unidade	Ator	Referência
Indicador: Sustentabilidade – Característica: Heterogeneidade									
1	Social	Sustentabilidade	Distribuição Geográfica	Númro de países que têm parceiros na comunidade	$X = N$ N = Número de países onde existem colaboradores	$X > P$ P = Parâmetro Quanto maior o número de países melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
2	Técnica	Sustentabilidade	Proximidade Semântica	Valor da médias da proximidade semântica dos serviços do ECOS	$X = N$ N = Média da proximidade semântica	$X > P$ P = Parâmetro Quanto maior o valor da média melhor	C –Valor da proximidade semântica	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
3	Técnica	Sustentabilidade	Tipos de Nós	Número dos tipos diferentes de nós presentes na rede de interação	$X = N$ N = Número de tipos diferentes de nós	$X > P$ P = Parâmetro Quanto maior o número de tipos melhor	Un - Unidade	Gerente de Rede	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
Indicador: Sustentabilidade – Característica: Habilidade de Regeneração									
4	Social	Sustentabilidade	Tempo de Trabalho Conjunto	Esforço conjunto dos membros da comunidade	$X = \sum^1_N T$ T = Horas trabalhadas por dia N = Numero de dias de esforço conjunto	$X > P$ P = Parâmetro Quanto maior o número de horas melhor	H - Horas	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
5	Social	Sustentabilidade	Novos Membros	Número de novos membros adicionados à comunidade em determinado número de dias	$X = \sum^1_N T$ T = Número de novos membros por dia N = Número de dias	$X > P$ P = Parâmetro Quanto maior o número de novos melhor	Un - Unidade	Colaborador	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
Indicador: Sustentabilidade – Característica: Balanceamento de Esforço									
6	Social	Sustentabilidade	Número de Commits	Numero total de commits feitos pelos desenvolvedores	$X = (\sum^1_N) \sum^1_A T$ T = Número de commits A = Número de desenvolvedores N = Número de dias	$X > P$ P = Parâmetro Grande número de commits pode indicar retrabalho	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
7	Social	Sustentabilidade	Membros Ativos	Número de comunidades com pelo menos um membro ativo member	$X = T$ T = Número de comunidades com membros ativos	$X > P$ P = Parâmetro Quanto maior o número de comunidades melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)

8	Social	Sustentabilidade	Participatividade	Número de comunidades com desenvolvedores ativos	$X = T$ T = Número de comunidades com desenvolvedores ativos	$X > P$ P = Parâmetro Quanto maior o número de comunidades melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
9	Social	Sustentabilidade	Esfroço total	Montante total de esforço conjunto de todos os membros da comunidade	$X = \sum^1_N T$ T = Número de horas trabalhadas por desenvolvedor N = Número de desenvolvedores	$X > P$ P = Parâmetro Quanto maior o número de horas melhor	H - Horas	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
Indicador: Sustentabilidade – Característica: Expertise									
10	Social	Sustentabilidade	Participação em Eventos	Número de participantes nos eventos da comunidade	$X = T$ T = Número de participantes	$X > P$ P = Parâmetro Quanto maior o número de participantes melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
Indicador: Sustentabilidade – Característica: Visibilidade									
11	Social	Sustentabilidade	Divulgação de Tarefas	Número de notificações de tarefas divulgados	$X = T$ T = Número de notificação de tarefas	$X > P$ P = Parâmetro Quanto maior o número de tarefas melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
12	Técnica	Sustentabilidade	Downloads Feitos	Número de downloads da plataforma feitos do site principal ou de links alternativos	$X = \sum^1_N T$ T = Número mensal de downloads N = Número de meses	$X > P$ P = Parâmetro Quanto maior o número de downloads melhor	Un - Unidade	Gerente de Comunicações	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
13	Social	Sustentabilidade	Número de Leitores	Número de leitores na comunidade	$X = T$ T = Número de leitores	$X > P$ P = Parâmetro Quanto maior o número de leitores melhor	Un - Unidade	Gerente de Comunicações	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
14	Social	Sustentabilidade	Publicações Científicas	Número de publicações científicas	$X = T$ T = Número de publicações científicas	$X > P$ P = Parâmetro Quanto maior o número de publicações científicas melhor	Un - Unidade	Gerente de Comunicações	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
15	Social	Sustentabilidade	Citações em Mídias Sociais e Blogs	Número de citações da comunidade em Mídias Sociais e Blogs	$X = T$ T = Número total de citações	$X > P$ P = Parâmetro Quanto maior o número de citações melhor	Un - Unidade	Usuário	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)

16	Social	Sustentabilidade	Requisição De Acesso à Página Inicial	Número de requisições recebidas pela pagina inicial da plataforma	$X = T$ T = Número total de requisições recebidas	$X > P$ P = Parâmetro Quanto maior o número de requisições melhor	Un - Unidade	Gerente de Comunicações	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
Indicador: Diversidade									
16	Social	Diversidade	Número de Desenvolvedores	Númro total de desenvolvedores na comunidade	$X = T$ T = Número de desenvolvedores	$X > P$ P = Parâmetro Quanto maior o número de desenvolvedores melhor	Un - Unidade	Gerente de Projetos	(Dhungana et al. 2010) (Amorim et al. 2017)
18	Social	Diversidade	Grupos de Usuários	Número total de grupos de usuários na comunidade	$X = T$ T = Número de grupos de usuários	$X > P$ P = Parâmetro Quanto maior o número de grupos melhor	Un - Unidade	Usuário	(Dhungana et al. 2010) (Amorim et al. 2017)
18	Técnica	Diversidade	Linguagens de Programação Suportadas	Número de linguagens de programação suportadas pela plataforma	$X = T$ T = Número de linguagens de programação	$X > P$ P = Parâmetro Quanto maior o número de linguagens suportadas melhor	Un - Unidade	Gerente de Projetos	(Dhungana et al. 2010) (Amorim et al. 2017)
20	Negócio	Diversidade	Plano para Colapso	Existencia de um plano contra o colapso do ECOS	$X = T$ T = Existência de plano contra colapso	$X = Y N$ Desejável que seja positivo	-	Gerente de Projetos	(Dhungana et al. 2010) (Bosch 2017)
Indicador: Produtividade									
21	Social	Produtividade	Novos Projetos	Número de projetos adicionados	$X = \sum^1_N T$ T = Número de projetos N = Número de meses	$X > P$ P = Parâmetro Quanto maior o número de novos projetos melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014) (Bosch 2017)
22	Social	Produtividade	Eventos da Comunidade	Eventos na comunidade	$X = \sum^1_N T$ T = Número de eventos N = Número de meses	$X > P$ P = Parâmetro Quanto maior o número de eventos melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014) (Amorim et al. 2017)
23	Técnica	Produtividade	Artefatos Adicionados	Número de artefatos adicionados ao repositório	$X = \sum^1_N T$ T = Número de artefatos N = Número de dias	$X > P$ P = Parâmetro Quanto maior o número de artefatos melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014) (Amorim et al. 2017)
24	Social	Produtividade	Mensagens Transmitidas	Número de mensagens transmitidas pelos	$X = \sum^1_N T$ T = Número de mensagens	$X > P$ P = Parâmetro Quanto maior o número de mensagens	Un - Unidade	Colaborador	(Jansen 2014) (Amorim et al. 2017)

				canais de comunicação	N = Número de dias	transmitidas melhor			
25	Técnica	Produtividade	Média de Tempo para Correção de Erros	Média de tempo necessário para correção de um erro de código	$X = T$ T = Número de horas para fixar o erro	$X \leq P$ P = Parâmetro Quanto menor a média de horas melhor	H - Horas	Gerente de Projetos	(Jansen 2014) (Amorim et al. 2017)
26	Social	Produtividade	Parceiros Adicionados	Número de novos parceiros adicionados	$X = \sum^1_N T$ T = Número de parceiros N = Número de dias	$X > P$ P = Parâmetro Quanto maior o número de parceiros melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014) (Amorim et al. 2017)
27	Social	Produtividade	Número de Usuários	Número total de usuários	$X = \sum^1_N T$ T = Número de usuários N = Número de dias	$X > P$ P = Parâmetro Quanto maior o número de usuários melhor	Un - Unidade	Usuário	(Jansen 2014) (Amorim et al. 2017)
28	Social	Produtividade	Tempo Médio de Uso	Média de tempo de uso da plataforma pelos usuários	$X = (\sum^1_N T)/N$ T = Número de horas de uso N = Número de dias	$X > P$ P = Parâmetro Quanto maior a média de tempo de uso melhor	H - Horas	Usuário	(Jansen 2014) (Amorim et al. 2017)
Indicador: Robustez – Característica: Interrelação									
29	Técnica	Robustez	Número de Conexões	Maior número de conexões entre os nós da rede de interação	$X = T$ T = Número de conexões	$X > P$ P = Parâmetro Quanto maior o número de conexões melhor	Un - Unidade	Gerente de Rede	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
30	Técnica	Robustez	Capacidade de Conexões	Capacidade máxima de conexões de rede	$X = T$ T = Capacidade de conexões	$X > P$ P = Parâmetro Quanto maior a capacidade melhor	Un - Unidade	Gerente de Rede	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
31	Técnica	Robustez	Razão entre Número de Conexões e Capacidade de Conexão	Razão entre o maior número de conexões e a capacidade de conexão da rede	$X = T/N$ T = Número máximo de conexões N = Capacidade de conexões	$X \leq P$ P = Parâmetro Quanto menor o valor da razão melhor	Un - Unidade	Gerente de Rede	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
32	Técnica	Robustez	Centralidade de Nós	Número máximo de conexões de nó de rede com o maior número de conexões	$X = T$ T = Número de conexões	$X > P$ P = Parâmetro Quanto maior a capacidade melhor	Un - Unidade	Gerente de Rede	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)

33	Negócio	Robustez	Parceiros Externos	Número de parceiros externos na plataforma	$X = T$ T = Número de parceiros	$X > P$ P = Parâmetro Quanto maior o número de parceiros melhor	Un - Unidade	Gerente Financeiro	(Franco-Bedoya et al. 2014) (Amorim et al. 2017) (Bosch 2017)
Indicador: Robustez – Característica: Agrupamento									
34	Social	Robustez	Tipos de Produtos	Número de tipos de produtos presentes na plataforma	$X = T$ T = Número de tipos de produtos	$X > P$ P = Parâmetro Quanto maior o número de tipos melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
35	Social	Robustez	Máxima Colaboração	Maior tempo de contribuição da comunidade para um projeto	$X = T$ T = Horas de contribuição	$X > P$ P = Parâmetro Quanto maior o número de horas melhor	H - Horas	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
36	Social	Robustez	Projetos Ativos	Número de projetos ativos	$X = T$ T = Número de projetos	$X > P$ P = Parâmetro Quanto maior o número de projetos melhor	Un - Unidade	Gerente de Projetos	(Franco-Bedoya et al. 2014) (Amorim et al. 2017)
Indicador: Robustez – Característica: Consistência Financeira									
37	Negócio	Robustez	Número de Parceiros	Número total de parceiros que a plataforma possui	$X = T$ T = Número de parceiros	$X > P$ P = Parâmetro Quanto maior o número de parceiros melhor	Un - Unidade	Gerente Financeiro	(Jansen 2014) (Bosch 2017)
38	Negócio	Robustez	Patrocínio Comercial	Número de patrocinadores comerciais que a plataforma possui	$X = T$ T = Número de patrocinadores	$X > P$ P = Parâmetro Quanto maior o número de patrocinadores melhor	Un - Unidade	Gerente Financeiro	(Jansen 2014) (Amorim et al. 2017) (Bosch 2017)
39	Negócio	Robustez	Valor Total de Contribuições	Valor total de contribuição recebido pela plataforma	$X = T$ T = Valor de contribuição	$X > P$ P = Parâmetro Quanto maior o valor de contribuição melhor	US\$ - Dollar R\$ - Real	Gerente Financeiro	(Jansen 2014) (Amorim et al. 2017) (Bosch 2017)
40	Negócio	Robustez	Contribuintes Ativos	Número de contribuintes ativos na plataforma	$X = T$ T = Número de contribuintes	$X > P$ P = Parâmetro Quanto maior o número de contribuintes melhor	Un - Unidade	Gerente Financeiro	(Jansen 2014) (Amorim et al. 2017) (Bosch 2017)

41	Social	Robustez	Usuários Frequentes	Número de usuários frequentes na plataforma	$X = T$ T = Número de usuários	$X > P$ P = Parâmetro Quanto maior o número de usuários melhor	Un - Unidade	Usuário	(Jansen 2014) (Amorim et al. 2017)
Indicador: Criação de Nicho									
42	Técnica	Criação de Nicho	Documentação	Presença de documentação na plataforma	$X = T$ T = Existência de documentação	$X = Y N$ Desejável que seja positivo	-	Gerente de Projetos	(Jansen 2014)
43	Negócio	Criação de Nicho	Tipos de Contribuintes	Número de tipos de contribuintes na comunidade	$X = T$ T = Número de tipos de contribuintes	$X > P$ P = Parâmetro Quanto maior o número de contribuintes melhor	Un - Unidade	Gerente Financeiro	(Jansen 2014) (Bosch 2017)
44	Técnica	Criação de Nicho	Tipos de Projetos de Aplicação	Número de tipos de projetos de aplicação desenvolvidos pela comunidade	$X = T$ T = Número de tipos de projetos	$X > P$ P = Parâmetro Quanto maior o número de tipos melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014)
45	Técnica	Criação de Nicho	Suporte à Linguagens Naturais	Suporte de linguagens naturais pela plataforma	$X = T$ T = Suporte a linguagem natural	$X = Y N$ Desejável que seja positivo	-	Gerente de Projetos	(Jansen 2014)
46	Técnica	Criação de Nicho	Tecnologias Suportadas	Número de tipos de tecnologias suportadas pela plataforma	$X = T$ T = Número de tecnologias suportadas	$X > P$ P = Parâmetro Quanto maior o número de tecnologias suportadas melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014)
47	Técnica	Criação de Nicho	Tecnologias de Desenvolvimento Suportadas	Número de tipos de tecnologias de desenvolvimento suportadas pela plataforma	$X = T$ T = Número de tecnologias de desenvolvimento suportadas	$X > P$ P = Parâmetro Quanto maior o número de tecnologias de desenvolvimento suportadas melhor	Un - Unidade	Gerente de Projetos	(Jansen 2014)

Tabela 2: Métricas de saúde da arquitetura Heal Me.

3.2. ONTOHEALTH

Em uma versão inicial da ontologia, especificada antes do estudo observacional relatado na seção 3.1, cinquenta e oito métricas foram geradas e incluídas na ontologia, a partir do mapeamento sistemático relatado no capítulo 2. As métricas para o indicador de sustentabilidade foram extraídas de (Franco-Bedoya et al. 2016) e (Amorim et al. 2017). As métricas para o indicador de diversidade foram extraídas de (Dhungana et al. 2010) e (Amorim et al. 2017). As métricas para definir os indicadores de produtividade e robustez (Franco-Bedoya et al. 2014), (Amorim et al. 2017) e (Bosch 2017). Por fim, as métricas para definir o indicador criação de nicho foram extraídas de (Jansen 2014) e (Bosch 2017).

A partir da definição das métricas agrupadas pelo mapeamento sistemático e pelo estudo observacional, uma ontologia, denominada *OntoHealth* foi especificada com o intuito de formalizar as métricas capturadas e realizar novas conexões entre os dados de um ECOS, de forma a capturar informações adicionais para a análise da saúde. Essa captura de informações adicionais se dá através de regras ontológicas definidas na ontologia e processamento de máquinas de inferência de forma a descobrir novas relações entre os mesmos.

A *OntoHealth* modela o domínio de ECOS, sua rede de colaboradores, atores, aplicativos agregados à plataforma e às keystones. A Figura 10 apresenta a visão geral da *OntoHealth*. As classes que compõem a *OntoHealth* são:

- **Companhias de desenvolvimento:** Esta classe modela as empresas envolvidas no desenvolvimento dos aplicativos que compõem o ECOS. Esta possui a subclasse *keystone*. Para ser classificada como *keystone*, a companhia deve ser mantenedora da plataforma;
- **Produto de software:** Esta classe modela os produtos de software componentes do ECOS, e que são produzidos pelas companhias de desenvolvimento. Um produto é caracterizado na subclasse plataforma caso seja a plataforma do ECOS;
- **Desenvolvedor:** Essa classe modela os desenvolvedores que são ligados ao ECOS. Estes desenvolvedores podem ser internos, ligados a uma empresa de desenvolvimento, ou externos, independentes. Estes podem estar agrupados também em comunidades ou grupos de desenvolvimento;

- Comunidade: esta classe agrupa desenvolvedores e empresas de desenvolvimento, modelando a comunidade do ECOS;
- Eventos: modela os eventos que ocorrem na comunidade;
- Projetos: modela os projetos vinculados à comunidade, tanto internos quanto externos à plataforma;
- Tecnologia: modela todas as tecnologias e linguagens de desenvolvimento utilizadas no ECOS.

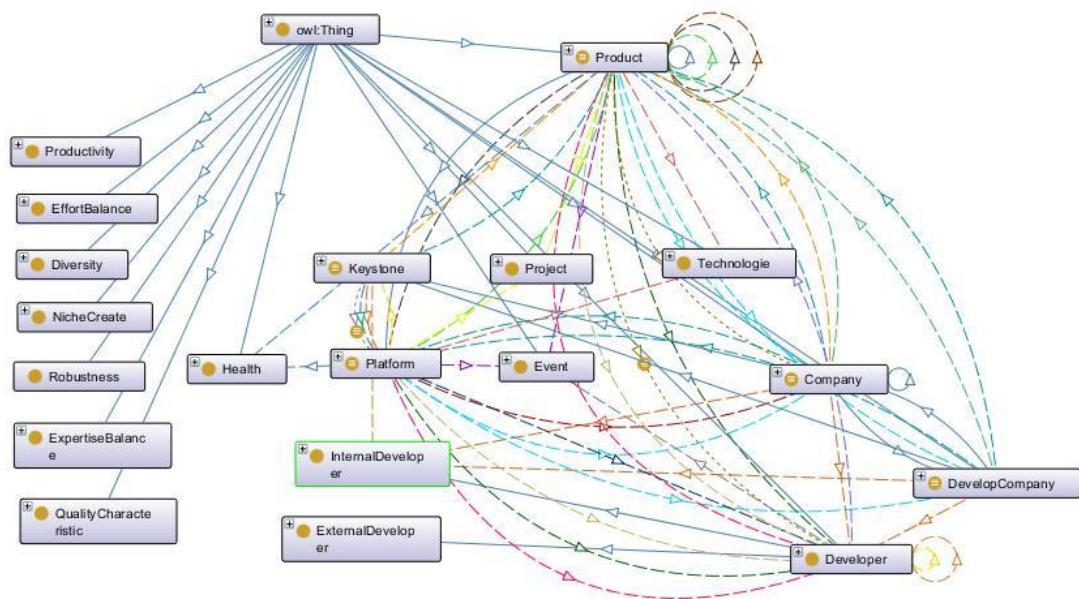


Figura 10: Visão geral da ontologia *OntoHealth*.

Para modelar algumas métricas, são utilizados dados (*dataproperty*). Tem-se como exemplo deste tipo de métrica as métricas *plano para colapso* e *documentação*, que utilizam dados lógicos (booleanos) para definição da métrica. A métrica *tempo para correção de erros* utiliza dados numéricos. A Figura 11 apresenta as *dataproperties* da ontologia.

- **ActiveDeveloper**
- **activeDeveloperPar**
- **activeMembers**
- **activeMembersPar**
- **averageTimeUse**
- **averageTimeUsePar**
- **bugFixTime**
- **bugFixTimePar**
- **ConnectivityCapacity**
- **ConnectivityCapacityPar**
- **createData**
- **createDataPar**
- **DeveloperCommits**
- **developerCommitsPar**
- **DeveloperHoursWork**
- **developerHoursWorkPar**
- **existPlanForCollapse**
- **existPlanForCollapsePar**
- **GreaterCollaboration**
- **GreaterCollaborationPar**
- **hasProgrammingExperienceTimeValue**
- **haveDocumentation**
- **haveDocumentationPar**
- **nodesCentrality**

Figura 11: *OntoHealth DataProperty*.

Para o relacionamento entre as classes, são definidas propriedades de objetos (*objectproperty*). Estas propriedades fazem as ligações das classes, permitindo a inferência de conhecimento através destas ligações. A Figura 12 ilustra estas propriedades.

- **hasActiveDeveloper**
- **hasCertificatedPlatformProperty**
- **hasCertificatedProductProperty**
- **hasCertified**
- **hasComposed**
- **hasContributedWithCommits**
- **hasContributedWithFiles**
- **hasContributedWithHours**
- **hasContributedWithReleases**
- **hasDeveloped**
- **hasEvent**
- **hasExperience**
- **hasExperienceTimeProperty**
- **hasFile**
- **hasGreatEvent**
- **hasGreatWorkTogether**
- **hasHeterogeneity**
- **hasLinkTo**
- **hasLive**
- **hasLiveDifferentCountry**
- **hasMaintained**
- **hasModificated**
- **hasNetworkNode**
- **hasNewMember**
- **hasParameter**

Figura 12: *OntoHealth ObjectProperty*.

Outra forma importante para descoberta de conhecimento existente na ontologia *OntoHealth* são as *property chains*. Estas propriedades são capazes de analisar cadeias de propriedades de objetos, e efetuar inferências entres os indivíduos envolvidos nestas. Como exemplo, pode-se citar a definição de companhias de desenvolvimento como *keystone*. Caso o indivíduo presente na ontologia for uma empresa de desenvolvimento, possuir um produto de software, e este for uma plataforma de ECOS, a empresa é uma *keystone*. A Figura 13 ilustra a construção da *property chain isKeystone*. As Figuras 11, 12 e 13 tem o intuito de ilustrar os componentes utilizados na descoberta de conhecimento da *OntoHealth*.



Figura 13: Property Chain isKeystone.

Ao instanciar essa ontologia com dados coletados do ECOS e usando regras expressas em *Semantic Web Rules Language* (SWRL) (Horrocks et al. 2004), é possível apresentar uma visão geral inicial da saúde da plataforma, avaliando suas características e indicadores.

Para realizar as avaliações, a ontologia utiliza 19 regras SWRL que processam as métricas disponíveis na literatura para analisar cada um dos indicadores de saúde e suas respectivas características. A próxima subseção descreverá as métricas utilizadas. A Figura 14 apresenta as métricas modeladas em linguagem SWRL.

```
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfCountries(?p, ?b) ^ numberOfCountriesPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ semanticClosenessAvg(?p, ?d) ...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ timeWorkTogether(?p, ?b) ^ timeWorkTogetherPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ numberOfNewMembers(?p, ?d) ^ ...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ DeveloperCommits(?p, ?b) ^ developerCommitsPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ activeMembers(?p, ?f) ^ activeM...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfEventParticipants(?p, ?f) ^ numberOfEventParticipantsPar(?a, ?g) ^ swrlb:greaterThan(?f, ?g) -> ExpertiseBalanc...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfJobAdvertisements(?p, ?d) ^ numberOfJobAdvertisementsPar(?a, ?e) ^ swrlb:greaterThan(?d, ?e) ^ numberOfD...
ExpertiseBalance(?x) ^ Visibility(?x) ^ Platform(?x) ^ RegenerationAbility(?x) ^ Heterogeneity(?x) ^ EffortBalance(?x) -> Sustainability(?x)
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfDevelopers(?p, ?n) ^ numberOfDevelopersPar(?a, ?d) ^ swrlb:greaterThan(?n, ?d) ^ numberOfUsersGroups(?p, ?...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfProjectsAdded(?p, ?b) ^ numberOfProjectsAddedPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ numberOfEvents(?p, ...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfNodesConnections(?p, ?b) ^ numberOfNodesConnectionsPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ Connectivit...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfProductTypes(?p, ?b) ^ numberOfProductTypesPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ GreaterCollaboration(?...
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ numberOfPartners(?p, ?b) ^ numberOfPartnersPar(?a, ?c) ^ swrlb:greaterThan(?b, ?c) ^ numberOfCommercialSponsors(?...
Platform(?p) ^ Interrelatedness(?p) ^ Clustering(?p) ^ FinancialConsistence(?p) -> Robustness(?p)
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^ haveDocumentation(?p, ?d) ^ haveDocumentationPar(?a, ?h) ^ swrlb:equal(?d, ?h) ^ numberOfContributorsTypes(?p, ?n) ...
Robustness(?x) ^ NicheCreate(?x) ^ Platform(?x) ^ Sustainability(?x) ^ Diversity(?x) ^ Productivity(?x) -> Health(?x)
DeveloperCommits(?x, ?d) ^ swrlb:greaterThan(?d, 19) ^ isDevelopedOf(?x, ?y) -> hasContributedWithCommits(?x, ?y)
swrlb:greaterThan(?d, 19) ^ DeveloperHoursWork(?y, ?d) ^ isDevelopedOf(?x, ?y) -> hasContributedWithHours(?x, ?y)
swrlb:equal(?d, true) ^ ActiveDeveloper(?y, ?d) ^ isDevelopedOf(?x, ?y) -> hasActiveDeveloper(?x, ?y)
hasContributedWithCommits(?x, ?y) ^ hasContributedWithReleases(?x, ?y) ^ hasActiveDeveloper(?x, ?y) ^ Platform(?x) ^ hasContributedWithHours(?x, ?y) ^ hasGreatWorkToge...
hasParticipated(?y, ?k) ^ isDevelopedOf(?x, ?y) ^ hasEvent(?x, ?k) -> hasParticipativeMembers(?x, ?y)
```

Figura 14: Métricas modeladas em regras SWRL.

O indicador de sustentabilidade, por exemplo, tem cinco características: heterogeneidade, capacidade de regeneração, equilíbrio de esforços, expertise e visibilidade. O indicador de robustez possui três características, sendo elas: inter-relação, agrupamento e

consistência financeira. A análise de métricas define os indicadores diversidade, produtividade e criação de nicho.

Para exemplificar o uso da ontologia, pode-se utilizar a análise de saúde do ECOS Wordpress (Lingen et al. 2013). Os dados deste ECOS foram pesquisados em repositórios de desenvolvimento (por exemplo, GitHub) e comunidades. Estes dados foram instanciados na ontologia para efetuar as inferências de saúde. Foram instanciados 1.571 desenvolvedores na ontologia, para avaliar a métrica *número de desenvolvedores*. A ontologia avalia cada desenvolvedor de acordo com a comunidade que este está presente caso seja desenvolvedor externo, ou a empresa de desenvolvimento caso seja desenvolvedor interno. A inferência classifica cada desenvolvedor como pertencente à plataforma através destes dados. Por fim, este número é comparado com o parâmetro informado para esta métrica. Foi informado o parâmetro de 1500 desenvolvedores para esta métrica ser atendida. Como o número de desenvolvedores é maior que o parâmetro, a métrica foi atendida. Esta métrica é parte da definição do indicador diversidade. Após avaliação das outras métricas, é avaliada a presença de diversidade no ECOS através da regra SWRL apresentada na Figura15.

```
Platform(?p) ^ Parameter(?a) ^ hasParameter(?p, ?a) ^
numberOfDevelopers(?p, ?n) ^ numberOfDevelopersPar(?a, ?
d) ^ swrlb:greaterThan(?n, ?d) ^ numberOfUsersGroups(?p,
?u) ^ numberOfUsersGroupsPar(?a, ?g) ^ swrlb:greaterThan
(?u, ?g) ^ numberOfProgrammingLanguagesSupported(?p, ?l)
^ numberOfProgrammingLanguagesSupportedPar(?a, ?s) ^
swrlb:greaterThan(?l, ?s) ^ existPlanForCollapse(?p, ?f)
^ existPlanForCollapsePar(?a, ?o) ^ swrlb:equal(?f, ?o)
-> Diversity(?p)|
```

Figura15: SWRL para diversidade.

As regras SWRL são utilizadas na ontologia para definir as métricas, indicadores e características conforme o exemplo apresentado. Estas definições fazem parte do processo de avaliação da Arquitetura Heal Me, que será apresentado da próxima seção.

3.3. PROCESSO DE AVALIAÇÃO DA SAÚDE NA ARQUITETURA HEAL ME

Com base na validação de métricas através do estudo observacional e a definição das mesmas na ontologia OntoHealth, foi possível construir um processo formal de avaliação de saúde. Este norteia o comportamento das análises executadas pela arquitetura Heal Me. Desta forma,

é possível tanto a evolução da arquitetura, como o aumento de sua efetividade na avaliação da saúde do ECOS. Além disso, a formalização das métricas melhora a eficácia e a precisão dos resultados.

O agrupamento de métricas em dimensões de ECOS tem o potencial de mostrar o ambiente do ECOS, usando uma estrutura simples e realista. Como dito anteriormente, as métricas usadas pela arquitetura Heal Me foram automatizadas através de regras semânticas especificadas em linguagem SWRL através da ontologia *OntoHealth*. Portanto, os dados do ECOS a ser avaliados são instanciados na ontologia.

Alguns dados referentes às métricas podem ser capturados automaticamente pela arquitetura Heal Me, através da interação desta com o ECOS como uma aplicação integrada ao núcleo destes. Isso permite a alimentação automática de dados por meio do uso de APIs, simplificando o processo de obtenção dos dados. Outras métricas, que não permitem sua captura automática, podem ser fornecidas por avaliadores de domínio e especialistas, que interagem diretamente com a arquitetura, informando os parâmetros de avaliação individuais para cada métrica.

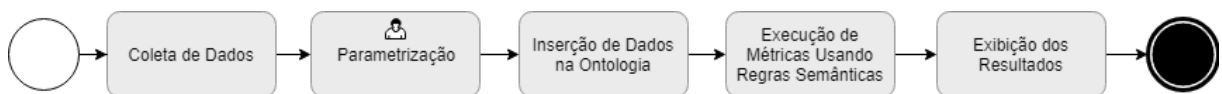


Figura 16: Processo de avaliação de saúde da arquitetura Heal Me.

O processo de avaliação desenvolvido para a arquitetura pode ser visto na Figura 16, no qual usa a notação do modelo BPMN (BPMN 2018). O processo de avaliação consiste em cinco passos:

- O primeiro passo é a coleta de dados. Os dados são coletados considerando o ambiente do ECOS. Estes estão relacionados ao desenvolvimento, usuários, rede de relacionamentos, entre outros. Esses dados são coletados automaticamente por meio de APIs específicas que se comunicam com repositórios existentes e/ou com aplicativos relevantes e com a plataforma. Cada dado coletado é avaliado por uma métrica.
- O segundo passo é a parametrização da análise. Estes parâmetros são informados por especialistas e variam de acordo com as características do ECOS. Os parâmetros são usados para avaliar se os dados coletados estão de acordo com a métrica.

- O terceiro passo é a instanciamento dos dados e parâmetros na ontologia. Esta etapa é necessária para a execução semi-automatizada da avaliação, através das regras semânticas (SWRL e *property chains*).
- O quarto passo é a execução das regras. Usando os parâmetros, as regras avaliam automaticamente se os dados estão de acordo ou não com a métrica.
- Finalmente, o quinto passo é a visualização dos resultados gerados pelo processo de avaliação. Nesta etapa, as técnicas de visualização são usadas para facilitar o entendimento do usuário. A visualização gráfica consegue apresentar de forma quantitativa o panorama de saúde apresentado pelo ECOS através da análise.

É importante ressaltar que a automação baseada em processos é muito importante devido ao grande número de métricas e à complexidade de executar a avaliação da saúde sem um suporte automático. As métricas formalizadas, aplicadas através do uso deste processo, podem aumentar a precisão dos resultados, trazendo mais confiança aos desenvolvedores e usuários do ECOS.

3.4. VISÃO GERAL DA ARQUITETURA HEAL ME

A arquitetura Heal Me tem como objetivo principal avaliar a saúde do ECOS através de métricas utilizando regras semânticas aplicadas aos seus dados. A análise inclui o ambiente do ECOS, seus produtos, redes de relacionamento, usuários e desenvolvedores. As métricas são aplicadas a esses dados, de acordo com cada um dos cinco indicadores de saúde. Com base nestes fatos e nos estudos descritos anteriormente neste capítulo, 47 métricas de avaliação da saúde foram definidas, detalhadas e apresentadas na Tabela 2 (Carvalho et al. 2017). Após o processamento destes dados é possível avaliar as condições de saúde da plataforma do ECOS, bem como suas restrições. A Figura 17 apresenta uma visão geral da arquitetura, que segue o modelo arquitetural em camadas e orientado a serviços (Buschmann et al. 2007).

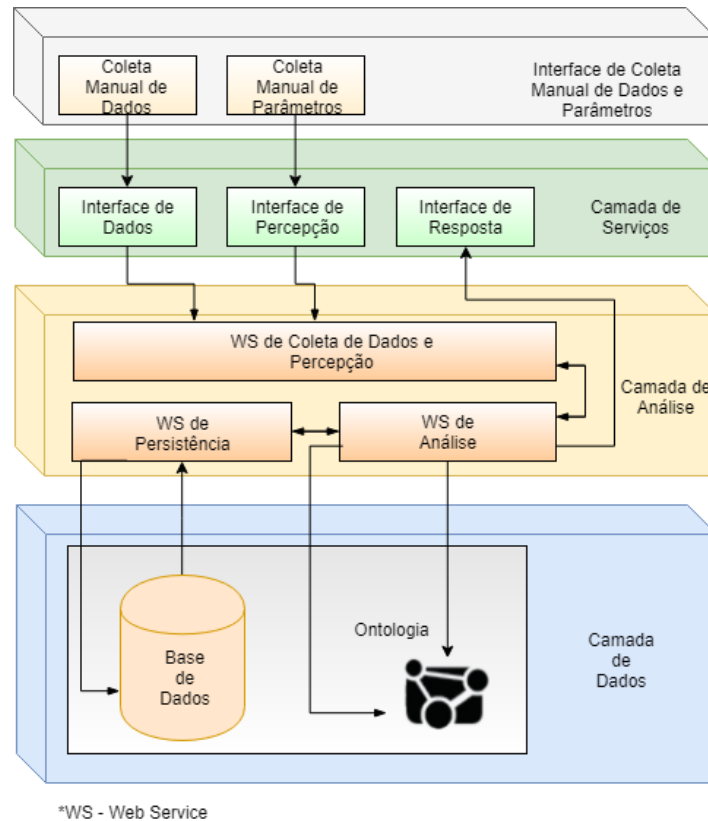


Figura 17. Visão geral da arquitetura Heal Me.

Conforme identificado no estudo observacional, a automação de processos referentes à saúde e qualidade de ECOS é uma preocupação recorrente. Diversos trabalhos propõem automação de análises e descoberta de conhecimento utilizando modelos e técnicas semânticas. Em acréscimo, também foi identificada a complexidade do ambiente e das tarefas para análise de saúde de ECOS. Com base nestes fatos, a arquitetura Heal Me foi desenvolvida como um software web, promovendo a automatização de processos em duas etapas:

- Automação da coleta de dados: Através da conexão com APIs de repositórios, os dados são coletados e cadastrados na base de dados para evitar o cadastro manual;
- Automação do processo de análises: onde sobre os dados cadastrados são aplicadas as métricas e obtidos os resultados através de regras semânticas. Devido à complexidade do ambiente e o volume de dados e métricas, tal atividade seria trabalhosa feita manualmente. Estas automações estão descritas no decorrer do texto.

A arquitetura é dividida em três camadas. A primeira, a camada de serviços, é responsável pelo acoplamento da arquitetura à estrutura do ECOS. Através de interfaces de comunicação, esta camada é responsável por capturar os dados dos componentes do ECOS. Estes incluem os dados que são modelados pela ontologia conforme descrito anteriormente. A interface de dados é responsável pela captura automática dos dados do ambiente de ECOS, conforme descrito anteriormente. Esta camada possui também a interface de coleta manual de dados e parâmetros, que é a interface web da arquitetura. Assim, nesta primeira versão, as interfaces de usuário para inserção de dados, dos parâmetros, e para geração dos relatórios das análises foram construídas como um *frontend web*.

Para conexão com o ambiente do ECOS e captura de seus dados, a arquitetura utiliza APIs de repositórios vinculados à plataforma, como a GitHub API (GitHub 2018). Por meio destas, os dados do projeto são capturados diretamente destes. Os dados que não estão presentes nos repositórios são capturados diretamente a partir do *frontend web*. Outra função importante desta interface é a captura de dados de percepção. Os usuários da plataforma fornecem esses dados para parametrizar as métricas, e viabilizando o processo de análise.

Na camada de análise estão os serviços responsáveis pelo fluxo de dados. Estes serviços capturam os dados provenientes das interfaces e, em seguida, os conectam ao serviço de análise. Este insere os dados coletados na ontologia, responsável pela sua interpretação. Esta camada inclui também a alimentação das interfaces, por meio das quais os dados analisados são disponíveis para tomadores de decisão, usuários e outros participantes do ECOS. Os dados analisados ficam disponíveis para serem consultados. Quando solicitado, o serviço de análise retorna os dados para a camada de serviços, para que estes relatórios possam ser gerados e exibidos. Já o serviço de persistência é responsável pela comunicação com o repositório de dados, armazenando e recuperando dados já processados. Esta camada, juntamente com a ontologia, é responsável pela automação do processo de análise.

A camada de dados engloba a ontologia, onde se encontram as regras de avaliação, e o repositório de dados. Este repositório armazena os dados da plataforma e seu ambiente, assim como os resultados obtidos do ECOS avaliado, já processados pela ontologia. Este cenário permite o uso futuro dos resultados por gerentes de projeto ou pela própria *keystone*. Este histórico é útil para comparação com novas avaliações de plataforma.

Conforme descrito pelo processo de avaliação presente na seção 3.3, a camada de serviços captura os dados da plataforma e dos parâmetros e os insere na ontologia. Utilizando estes parâmetros, as métricas avaliam os dados do ECOS, considerando as características e cada um dos indicadores de saúde. Por esta razão, é importante que o tomador de decisão que

utiliza a arquitetura tenha a percepção do ambiente ECOS e seus componentes. Para garantir que o resultado reflita a situação real de um ECOS é interessante que os gerentes de projeto ou desenvolvedores da *keystone* façam uso da arquitetura.

Com esses dados carregados na ontologia, o mecanismo de inferência é processado. Essa ação processa as regras usando os dados carregados e os parâmetros definidos, gerando as definições de saúde. Como resultado, o ECOS é classificado como tendo um indicador específico ou não. A presença dos cinco indicadores demonstra a saúde da plataforma, uma vez que estas são características mutuamente dependentes.

Por fim, o *frontend web* mostra os resultados do processo de avaliação e informa o status da plataforma de acordo com cada característica, indicador e finalmente exibe a avaliação geral de saúde.

3.4.1. TECNOLOGIAS UTILIZADAS

Os serviços componentes da arquitetura Heal Me foram desenvolvidos em linguagem Java (Java 2018), utilizando o padrão *RESTful* (Pautasso et al. 2008) e arquitetura de microserviços (Wu 2017). Para cada ação é disponibilizada um *endpoint*, identificado pelo verbo *Hypertext Transfer Protocol* (HTTP) correto. O carregamento de dados no *frontend Web* é efetuado por serviços do tipo GET. Para salvamento dos dados, são utilizados serviços do tipo POST alimentados por objetos *Extensible Markup Language* (XML).

Por fim, o tráfego dos dados entre o banco de dados e a ontologia, assim como a alimentação desta e execução de regras são feitas internamente nos serviços, utilizando a biblioteca ontológica Jena (Apache 2018). Através desta, os serviços conectam na ontologia, carregam os dados cadastrados na arquitetura, e disparam os processos de definições protagonizados pelas regras SWRL (Horrocks et al. 2004). Após, concluídos, os serviços retornam estes dados para a camada de serviços, alimentando os relatórios de análise.

O modelo de dados utilizado pela Camada de Dados é o modelo relacional (Elmasri e Navathe, 2005), que é acessado pelos serviços web utilizando o framework de persistência Hibernate (Hibernate 2018).

Como descrito, os serviços capturam os dados da base e instanciam a ontologia utilizando o framework Jena. Esta instanciação permite a execução das regras SWRL. As execuções e inferências são feitas utilizando o *reasoner* Pellet (Sirin et al. 2007).

Para um melhor entendimento do funcionamento da arquitetura Heal Me, detalhamos a seguir, um exemplo de uso da arquitetura.

3.5. EXEMPLO DE USO

A interface Web (*frontend web*) é responsável pela interação com os usuários. Através dela é possível a interação com a arquitetura, preenchimento de dados, captura automática de dados e visualização dos resultados de análises. Desta forma, apresentamos a seguir, a partir do exemplo do ECOS Wordpress (Wordpress 2018), um exemplo de uso e funcionamento da arquitetura.

Para alimentar os dados de entrada na arquitetura o usuário deve acessar a página *Platform Registration*. Esta possui um campo para cadastro do nome do ECOS a ser avaliado, e quarenta e sete campos, cada um referente à uma métrica usada na avaliação. Cada campo é identificado por nome e uma anotação do tipo *hint* para permitir ao usuário visualizar qual métrica que está sendo cadastrada. Como exemplo, para avaliar a métrica *número de países*, deve-se preencher este valor na tela de registro de plataforma. No exemplo do Wordpress, ele está presente em 120 países. Basta preencher o campo *Number of Countries* com o valor 120. Após alimentar todos os campos referentes às métricas, basta clicar em *Save*.

Figura18: Página *Platform Registration*.

Nesta página é possível também acessar a importação automática dos dados. Através da opção *AutoImport*, dados disponíveis nos repositórios do ECOS podem ser capturados para alimentar as análises. A opção de importação automática foi projetada para consumir APIs, conectado a estas através de serviços e consumindo seus dados.

Por este motivo, para captura dos dados de cada métrica é necessário informar um *endpoint* de serviço que proverá os dados. O processo consiste em conectar aos *endpoints* através de clientes web, capturar e converter os dados, caso necessário, e alimentar os objetos de domínio da arquitetura.

Metric	EndPoint
Active Members	
Number Of Communits	
Total Effor	
Number Of Downloads	

Import Close

Figura 19: Tela de importação automática de dados.

No entanto, as APIs disponíveis não possuem um formato padrão. Em acréscimo, os dados fornecidos não têm foco apenas no ambiente do ECOS, mas são generalizados para os projetos. Por causa destes formatos, para a captura automática dos dados de forma efetiva, é necessário o desenvolvimento de um conversor de dados. Este conversor é capaz de capturar os dados produzidos pelas APIs, modelá-los para o domínio de dados da arquitetura Heal Me e por fim, capturar os dados de interesse.

Para exemplificar, pode-se utilizar a métrica *número de leitores*. Para o exemplo de utilização do Wordpress, foi consumido o *endpoint*² de seu repositório de projetos do GitHub. O objeto de domínio deste *endpoint* é instanciado no serviço da arquitetura Heal Me, e captura o resultado do *endpoint*. O valor capturado foi de 30 leitores inscritos no projeto. Este valor foi capturado e alimentou o campo *Number of Readers* da interface de cadastro da plataforma.

² Disponível em: <https://api.github.com/repos/Wordpress/wordpress/subscribers>

Metric	EndPoint
Number Of Readers	https://api.github.com/repos/Wordpress/w
Number Of Communits	https://api.github.com/repos/Wordpress/w
Number Of Event Participants	https://api.github.com/repos/Wordpress/w
Number Of Downloads	https://api.github.com/repos/Wordpress/w

Import Close

Figura 20: Alimentando tela de captura automática.

Number of Readers:	30
Web page requests:	422.567

Figura 21: Dado cadastrado na interface de cadastro de plataformas.

Para a efetiva execução dos processos de análise, é necessária a correta parametrização da plataforma. Os parâmetros da arquitetura Heal Me são dados de referência, necessários para comparação dos dados do ECOS que foram alimentados na arquitetura. Para cada métrica é esperado um parâmetro. Através destes parâmetros, o usuário transmite à arquitetura sua perspectiva sobre o ambiente de ECOS que deseja analisar. Por isto, é indicado que estes parâmetros sejam cadastrados com base na experiência e vivência do usuário avaliador sobre ECOS, seus ambientes e complexidade.

Para cadastrar estes dados, a arquitetura Heal Me possui a página denominada *Parameter Registration*, apresentada na Figura 22. Semelhante à página de cadastro de plataformas, esta página apresenta um campo para cada métrica devidamente identificado. Para guiar os processos de cadastro, além de orientações sobre os parâmetros, e a forma correta de preenchimento, deve-se observar o campo Interpretação presente na Tabela 2. Seguindo o exemplo do Wordpress, avaliado pela arquitetura Heal Me, o parâmetro definido foi de 66 países. Semelhante ao preenchimento do dado, para informar o parâmetro basta preencher o campo *Number of Countries* com o valor de 66. Após preencher todos os parâmetros, deve-se clicar em *Save* para armazená-los.

Figura 22: Tela de cadastro de parâmetros.

A última etapa do processo de análise compreendido pela camada de serviços com o usuário é a visualização do resultado da análise. Para este fim, foi desenvolvido um relatório na página *Platform Analysis*. Após acessara seleção de plataformas, é disparado o processo de análise feito pela ontologia. Os resultados são apresentados nos gráficos da página.

A página de resultados da análise é dividida em três setores. Estes são responsáveis por exibir o grau que o ECOS atende a cada nível de granularidade dos componentes da análise. Esta granularidade é dividida em saúde geral, indicadores de saúde, características e métricas.

O primeiro setor é o *Analysis Overview*. O gráfico deste setor exibe a porcentagem atendida de cada indicador de saúde, a saber: sustentabilidade, diversidade, produtividade, robustez e criação de nicho. Em acréscimo, é exibido à esquerda o gráfico de saúde, apresentando a porcentagem total de saúde atingida. A média das porcentagens dos indicadores compõe a porcentagem apresentada por este gráfico. Este setor é apresentado na Figura 23.

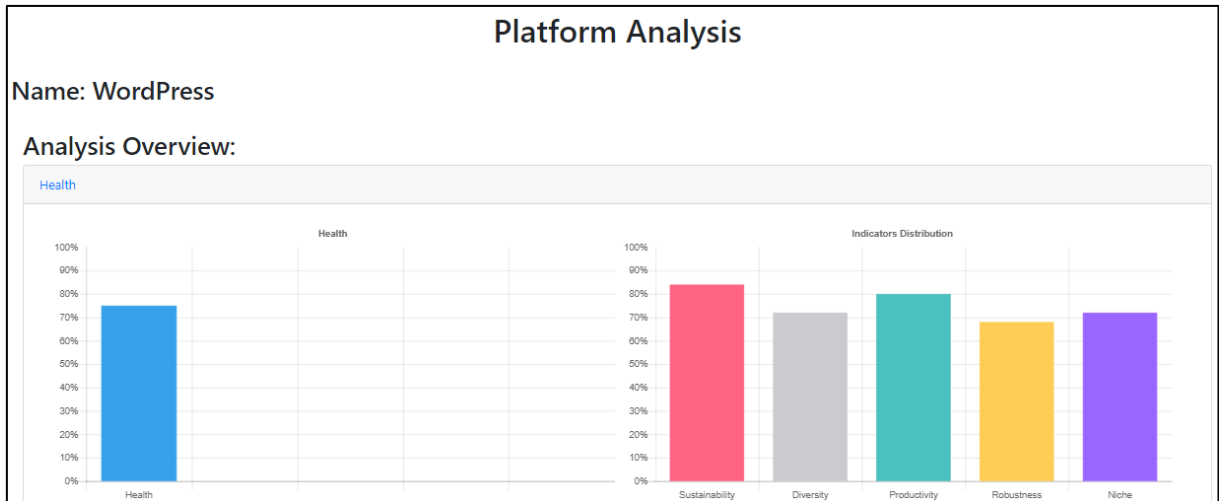


Figura 23: Health Overview.

O segundo setor é o *Individual Indicators*, ilustrado pela Figura 24. Este mostra a porcentagem das métricas atendidas que compõem cada uma das características de saúde individualmente. Estas características, por sua vez, compõem os indicadores. O indicador Sustentabilidade é composto pelas características Heterogeneidade, Habilidade de Regeneração, Balanceamento de Esforço, Balanceamento de Expertise e Visibilidade. Já o indicador Robustez é composto pelas características Interrelação, Consistência da Informação, Agrupamento e Consistência Financeira. Cada gráfico exibe a porcentagem de atendimento de seu indicador, calculado através da média das porcentagens de suas características. Os indicadores Diversidade, Produtividade e Criação de Nicho não são divididos em características. Por este motivo, para cada um destes é exibido um gráfico com a porcentagem de atendimento das métricas que os compõem.

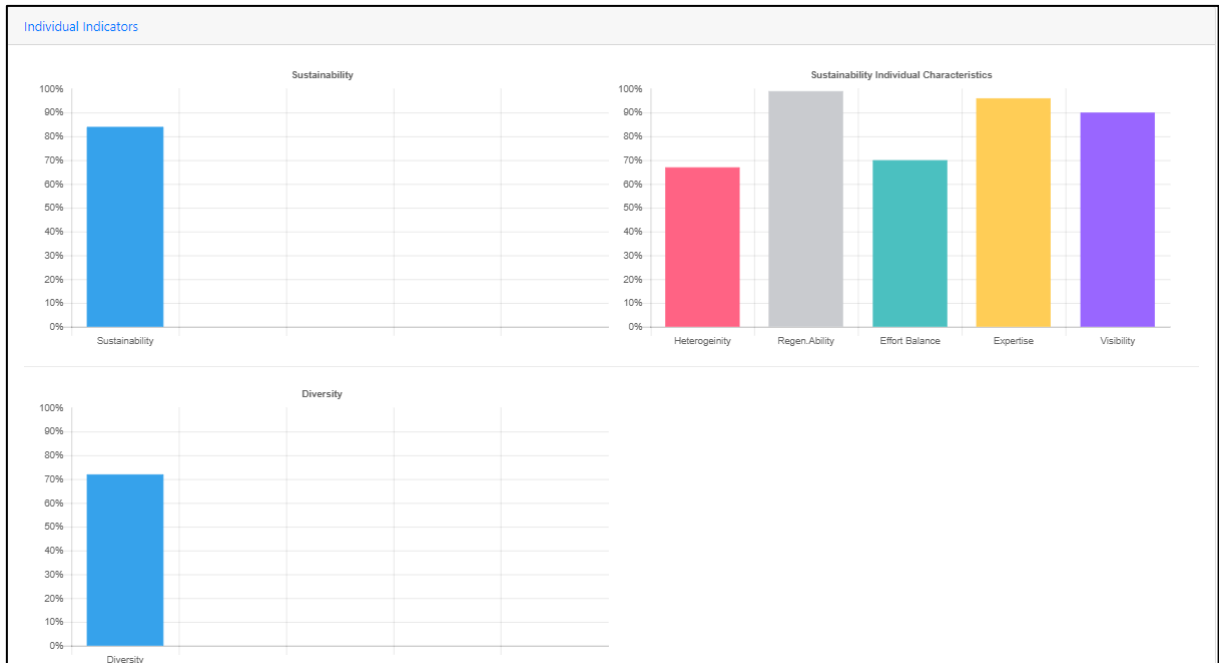


Figura 24: *Individual Indicators*.

Por fim, o ultimo setor mostra as métricas individualmente, agrupadas por característica ou indicador. As tabelas deste setor exibem o nome da métrica, o dado informado no cadastro da plataforma, o parâmetro correspondente e a porcentagem atingida pela medição.

O gráfico apresenta as métricas individualmente. Cada barra do gráfico representa uma métrica, e qual porcentagem a medição efetuada para aferi-la atingiu. No exemplo ilustrado na Figura 25 podemos visualizar a característica Heterogeneidade, que é composta pelas métricas Número de Países, Proximidade Semântica e Número de Tipos de Nós. No exemplo ilustrado, as métricas Número de Países e Número de Nós atingiram 100% da medição em comparação aos parâmetros informados. A métrica proximidade semântica não pode ser avaliada, pois não foram encontrados dados disponíveis sobre a plataforma Wordpress referentes à esta métrica. Com base nas definições da análise do Wordpress, seus dados e parâmetros, é possível visualizar que este ECOS atingiu 67% na característica Heterogeneidade.

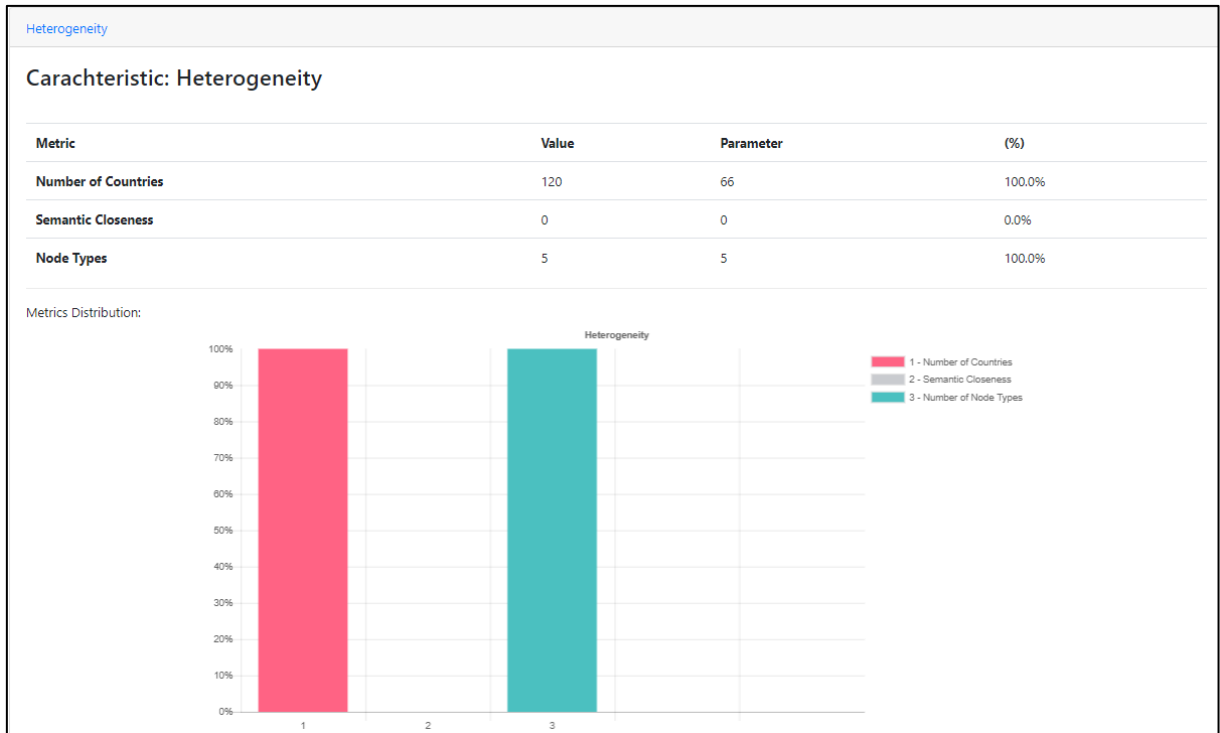


Figura 25: Gráfico individual da característica Heterogeneidade e suas respectivas métricas.

Os resultados das características atingidas pela plataforma Wordpress na análise efetuada pela arquitetura Heal Me estão descritos Tabela 3. A Tabela 4 descreve os resultados dos indicadores e o resultado da saúde.

Característica	Atingido (%)
Heterogeneidade	67
Habilidade de Regeneração	99
Balanceamento de esforço	70
Expertise	96
Visibilidade	90
Interrelação	25
Agrupamento	94
Consistência Financeira	85

Tabela 3: Resultado das características da plataforma Wordpress.

Indicador	Atingido (%)
Sustentabilidade	84
Diversidade	72
Produtividade	80
Robustez	68
Criação de Nicho	72
Total da Saúde	75

Tabela 4: Resultado dos indicadores e da saúde da plataforma Wordpress.

Além das páginas apresentadas e os componentes do processo de avaliação de saúde, a arquitetura Heal Me ainda disponibiliza um documento Guia de Uso³, para guiar o usuário para a correta utilização da arquitetura. Em acréscimo, a tabela de métricas apresentada na Tabela 2 também está disponível para facilitar a visualização e o entendimento das métricas.

3.6. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo descreveu os principais componentes da arquitetura Heal Me, a ontologia *OntoHealth*, o processo para seu desenvolvimento e um exemplo de uso da arquitetura. Este capítulo descreveu também o processo de validação das métricas componentes da arquitetura, a formalização efetuada para melhorar a precisão e facilitar a aplicação destas, e sua descrição em linguagem SWRL. Contudo, para obter indícios da efetividade da arquitetura foi necessária uma avaliação para visualizar a assertividade dos resultados produzidos e a eficiência das análises executadas. Essa avaliação foi executada por especialistas do domínio de ECOS que utilizaram a arquitetura. Esta foi feita no modelo estudo de caso, e é descrita no próximo capítulo.

³Disponível em: http://200.131.219.85:8080/HealMe/faces/files/Guia_de_uso_Heal_Me.pdf

4. AVALIAÇÃO DA PROPOSTA

Este capítulo apresenta o estudo de caso efetuado para avaliação da arquitetura Heal Me. Serão descritos os passos de planejamento, execução do estudo e coleta dos dados. Por fim, serão apresentados os resultados obtidos.

4.1. DEFINIÇÃO DO ESTUDO DE CASO

Para basear a utilização da estratégia do estudo de caso nesta avaliação, pode-se utilizar a definição de Yin (2015), onde este descreve que estudo de caso é indicado para pesquisas cujas questões são baseadas em “como” e “porque”, quando o pesquisador tem pouco controle sobre os eventos e quando os fenômenos ocorrem em situações reais. Apesar de outras estratégias de pesquisa basearem suas questões nos mesmos moldes do estudo de caso, este se baseia no pouco ou nenhum controle do pesquisador sobre o ambiente. Em acréscimo, o estudo de caso observa fatos contemporâneos, ou seja, que estão em acontecimento. Este cenário viabiliza o uso de estudo de caso no contexto desta pesquisa.

Outra realidade sobre os estudos de caso é que esta técnica é amplamente utilizada na engenharia de software. Segundo Runeson et al. (2012), um estudo de caso é uma investigação empírica que possui várias fontes de evidências, onde é analisado apenas um objeto de estudo. Este objeto e seus fenômenos devem atender ao requisito de contemporaneidade. Em acréscimo, em situações onde o limite entre os fenômenos e o contexto não é claro, é indicada a utilização de um estudo de caso. Com base neste cenário, foi identificada a necessidade de se aplicar um estudo de caso na avaliação desta pesquisa.

Para efetuar este estudo de caso, foram necessários alguns passos preparatórios. Após a arquitetura ter sido implementada como um software web, este foi publicado em um servidor aberto, onde seria possível o acesso por parte dos participantes. Após, foi criada a documentação de apoio que está descrita no planejamento. Para diminuir a complexidade e duração do processo de avaliação, fez-se necessário a disponibilidade de dados de ECOS reais para os participantes. Para isso foram pesquisados ECOS de destaque na área dentro da literatura. Com estes ECOS definidos, iniciou-se o processo de coleta de dados. Nos repositórios de cada ECOS, foram pesquisadas APIs com dados disponíveis referentes às métricas presentes na arquitetura Heal Me. Para os dados cujas APIs não estavam disponíveis

ou não foram localizadas, os dados foram coletados e cadastrados manualmente seguindo os passos descritos no Apêndice D. A descrição dos ECOS é feita a seguir.

Para este estudo de caso, foi proposto aos participantes efetuar a análise de três ECOS já conhecidos (Manikas 2016): KDE (KDE 2018), Eclipse (Eclipse 2018) e Wordpress (Wordpress 2018). Os três ECOS escolhidos possuem bastante destaque na comunidade acadêmica, e por este motivo foram eleitos para comporem o estudo de caso. O ECOS KDE (Amorim et al. 2017) possui mais de 20 anos de existência, e possui uma comunidade de desenvolvimento com mais de quatro mil colaboradores. Com foco em Open Source Software (OSS), é caracterizado pelo desenvolvimento de interfaces gráficas para *kernels* de sistemas, predominantemente sistemas Linux. Além de sua abertura e sua grande comunidade de colaboradores, outros fatos que caracterizam o KDE como ECOS é o grande número de eventos organizados pela comunidade, listas de discussão e sua presença em escala global.

O Eclipse (Berger et al. 2014) é uma comunidade de código aberto cujos projetos são concentrados na criação de uma plataforma de desenvolvimento aberta mantida pela Eclipse Foundation e que ajuda a cultivar uma comunidade de software livre e um ecossistema de produtos e serviços complementares. O Projeto Eclipse foi originalmente criado pela IBM em novembro de 2001 e suportado por um consórcio de fornecedores de software. Hoje, a comunidade do Eclipse é composta por pessoas e organizações do mercado de software.

Por fim, o Wordpress (Lingen et al. 2013) também reúne uma grande comunidade de desenvolvimento, onde muitos sites do tipo blog e até aplicações comerciais utilizam seus *plug-ins* na atualidade. Por possuir um núcleo aberto, seus *plug-ins* podem ser customizados para nichos específicos. Desta forma, a plataforma Wordpress tem atraído um número maior de usuários a cada dia, disponibilizando suporte aos usuários, assim como grande interação da comunidade.

Os dados utilizados para esta avaliação foram capturados das páginas das comunidades de cada ECOS⁴, do repositório OpenHub⁵, além dos repositórios dos projetos e da API fornecidos pelo GitHub⁶. Para geração dos parâmetros foi utilizada como base uma média dos dados dos ECOS cadastrados na arquitetura. Como cada ECOS possui características e enfoques diferentes, estes parâmetros foram analisados por especialistas para cada ECOS, e adaptados dentro das necessidades.

⁴Disponível em: <https://www.kde.org/>, <https://www.eclipse.org/> e <https://br.wordpress.com/>

⁵Disponível em: <https://www.openhub.net/>

⁶Disponível em: <https://developer.github.com/v3/>

4.2. CONDUÇÃO DA AVALIAÇÃO

4.2.1. PLANEJAMENTO

O primeiro passo para a execução do estudo foi o planejamento. O escopo do estudo de caso foi definido com base no método GQM (Basili et al. 1994): “**avaliar** o uso e funcionamento da arquitetura Heal Me com o **objetivo** de verificar a eficácia da análise de saúde e correto funcionamento da arquitetura Heal Me **do ponto de vista** de usuários especialistas **no contexto** de Ecossistemas de Software”.

Ainda para estabelecer os objetivos deste estudo, foram definidas duas questões de pesquisa. Estas questões foram derivadas diretamente da questão principal de pesquisa, foco deste trabalho:

- QP1: “A arquitetura Heal Me funciona corretamente? ”
- QP2: “As avaliações de saúde efetuadas pela arquitetura Heal Me são eficazes? ”

Com os objetivos definidos, pôde-se planejar a forma de execução. O estudo foi planejado para ser executado de forma online, sem interferência do pesquisador durante o processo. Para isso, foi disponibilizado para os participantes o endereço de acesso à arquitetura⁷. Em acréscimo, foi oferecido o Guia de Uso da plataforma, presente no Apêndice D, assim como a tabela de métricas utilizadas na arquitetura, apresentada na Tabela 2. Por fim, foi disponibilizado para os participantes o questionário dividido em três seções, presentes nos Apêndices A, B e C, respectivamente. A primeira seção apresenta o termo de consentimento para participação, onde são apresentadas todas as implicações de participar, ressaltando-se que a confidencialidade das identidades é mantida a todo momento, assim como a arbitrariedade da participação.

A segunda seção tem por objetivo caracterizar os participantes com relação à formação acadêmica, tempo de experiência acadêmica e profissional, experiência com ECOS e por fim, experiência com avaliação de saúde no contexto de ECOS. A última seção do questionário tem por objetivo capturar as experiências de uso e visão geral dos usuários sobre a arquitetura Heal Me. O questionário é composto por treze questões na escala de Likert (Likert 1932), para permitir maior precisão e visualização dos resultados. Em acréscimo, foram disponibilizadas três questões abertas, para os participantes apontarem pontos fortes e fracos da arquitetura, assim como sugestões de mudanças e melhorias para as próximas versões.

⁷Disponível em: <http://200.131.219.85:8080/HealMe/faces/index.xhtml>

Os dados coletados para a execução do estudo estão disponíveis para consulta⁸. Estes dados foram coletados do ambiente e dos repositórios dos ECOS com base nas métricas que compõem a arquitetura Heal Me. Parte destes foram coletados de forma automática, como os dados do ECOS Wordpress sobre número de leitores, citado anteriormente, capturado da API dos repositórios do GitHub deste ECOS. Foram encontrados 30 leitores no repositório. Outros dados foram capturados manualmente, como o dado sobre *projetos ativos*, do ECOS KDE, onde foram encontrados em seu portfólio, disponível na página da organização, 17 projetos ativos. Para esta avaliação, os dados não encontrados foram zerados. Juntamente com os dados, estão disponíveis os parâmetros utilizados.

Efetuada o planejamento do estudo, pode-se passar para a próxima etapa que é a execução.

4.2.2. EXECUÇÃO

A execução do estudo foi feita de forma online. As atividades ocorreram entre 22 de agosto e 1º de setembro de 2018. Os participantes acessaram a plataforma, utilizaram-na conforme orientado pelo Guia de Uso e responderam o questionário com base nas experiências vivenciadas. Estes responderam o questionário em dias e horários diferentes dentro o período citado. Como desempenho não é um fator crítico para o funcionamento da arquitetura, o tempo gasto na utilização e preenchimento do questionário não foi capturado. Após o prazo de participação, os participantes foram caracterizados e os resultados obtidos compilados. As próximas seções apresentarão estes resultados e as ameaças à validade.

4.2.3. CARACTERIZAÇÃO DOS PARTICIPANTES

Participaram do estudo dez indivíduos de diversas instituições de ensino e com experiência tanto acadêmica quanto na indústria de software. Através do questionário de caracterização, foi possível capturar a experiência dos participantes. A Figura 26 mostra a distribuição dos participantes dentro dos níveis de formação acadêmica. Pode-se observar que dentre estes, nove possuem grau de mestre e um possui nível de graduação.

⁸Disponível em: <https://goo.gl/wrtrMn>

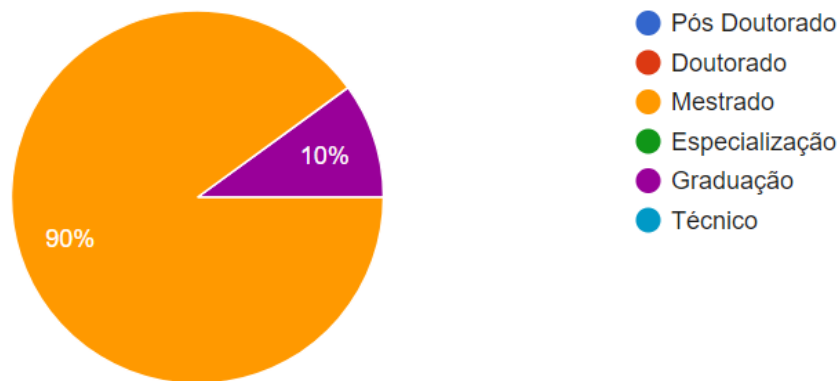


Figura 26: Grau acadêmico dos participantes.

A segunda caracterização efetuada foi quanto à atividade atual dos participantes. Nesta etapa, pode-se observar que três participantes atuam apenas na indústria, três atuam apenas na academia e quatro atuam em ambas as áreas. A Figura 27 apresenta esta distribuição.

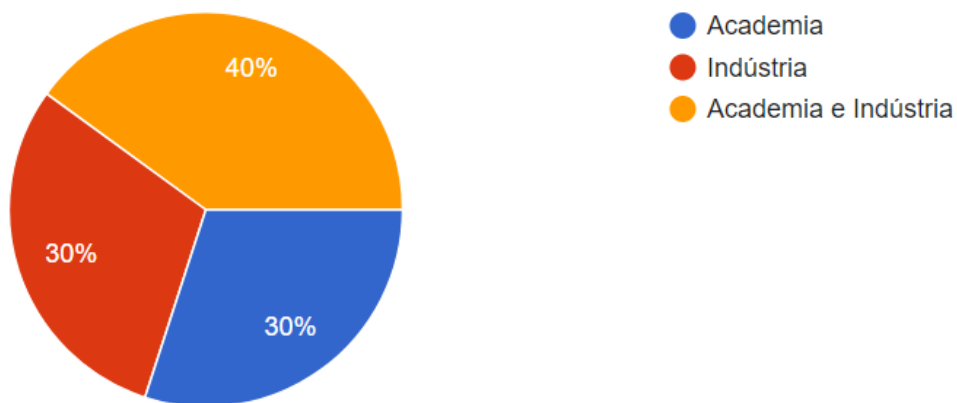


Figura 27: Distribuição dos participantes dentre as áreas de atuação.

Após caracterizar os participantes quanto ao grau acadêmico e a área de atuação, buscou-se avaliar a experiência destes. O primeiro passo para avaliar este quesito foi capturar o tempo de experiência dos participantes na academia e na indústria. A experiência acadêmica dos participantes varia de dois a vinte anos, enquanto o tempo de experiência na indústria varia de zero a vinte e quatro anos. Este cenário pode indicar que entre os participantes existem tanto pessoas com relativamente pouca experiência profissional na área de desenvolvimento de software, quanto pessoas com notória experiência e vivência.

Os quesitos seguintes foram:

- Experiência com ECOS, onde a distribuição das experiências varia de um a seis anos. Este cenário aponta que todos os participantes possuem o perfil mínimo para participar do estudo de caso, uma vez que são especialistas no domínio de interesse;

- Experiência como desenvolvedor/colaborador em ambientes de ECOS que variou de um a dez anos;
- Experiência dos participantes como gerente/responsável por processos em ECOS, que variou de zero a 3 anos, sendo que oito deles possuem pelo menos um ano;
- Experiência em definir arquitetura de plataforma/ambiente de ECOS, onde a variação foi de um a cinco anos;
- Experiência dos participantes com avaliação/controle de qualidade em ECOS, onde a variação foi de zero a dez anos, tendo sete participantes mais de um ano de experiência;
- Experiência dos participantes com avaliação/controle de saúde em ECOS, onde a variação foi de zero a dois anos, tendo cinco participantes mais de um ano de experiência.

A caracterização dos participantes é importante para indicar um maior grau de confiabilidade nas respostas obtidas, uma vez que os participantes estão de fato imersos no ambiente de ECOS e possuem experiência suficiente para efetuar uma análise correta. Os resultados desta avaliação são apresentados a seguir.

4.3. RESULTADOS

Os resultados apresentados nesta seção buscam responder os itens propostos no questionário de avaliação. Com base nestas respostas, busca-se atingir o objetivo da pesquisa, assim como responder as questões propostas. Foram dispostos treze itens na escala de Likert, distribuída de 1 a 5, onde 1 - discordo totalmente, 2 - discordo parcialmente, 3 - indiferente, 4 - concordo parcialmente e 5 - concordo totalmente.

O primeiro item apresentado no questionário foi “O panorama geral de saúde do ECOS foi facilmente compreendido na arquitetura Heal Me.” A Figura 28 apresenta a distribuição das respostas.

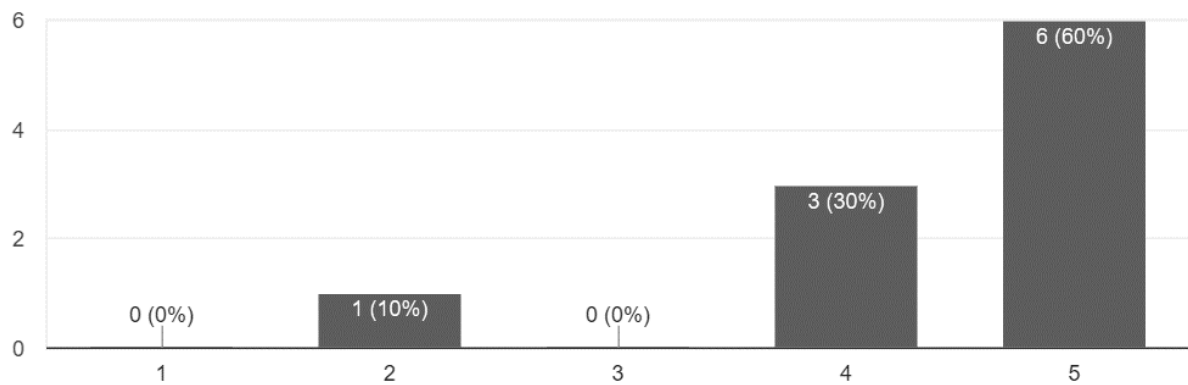


Figura 28: Resultado do item “O panorama geral de saúde do ECOS foi facilmente compreendido na arquitetura Heal Me.”

A distribuição das respostas, com maioria (30% Concordo Parcialmente e 60% Concordo Totalmente) concentrada nas opções concordo parcialmente e concordo totalmente, demonstram indícios de que a apresentação do panorama dos ECOS utilizados durante a avaliação foi compreendida pelos participantes. Contudo, a presença de um dos participantes ter avaliado como discordo parcialmente pode indicar que a compreensão do sistema da análise, composto por cinco indicadores, oito características e quarenta e sete métricas, não se apresenta de forma trivial. Com base nisto, para as próximas versões da arquitetura, será estudada uma apresentação mais compreensível dos componentes da análise.

O segundo item foi “Os níveis atingidos de saúde e de cada indicador individualmente foi facilmente visualizado na arquitetura Heal Me.” As respostas podem ser observadas na Figura 29.

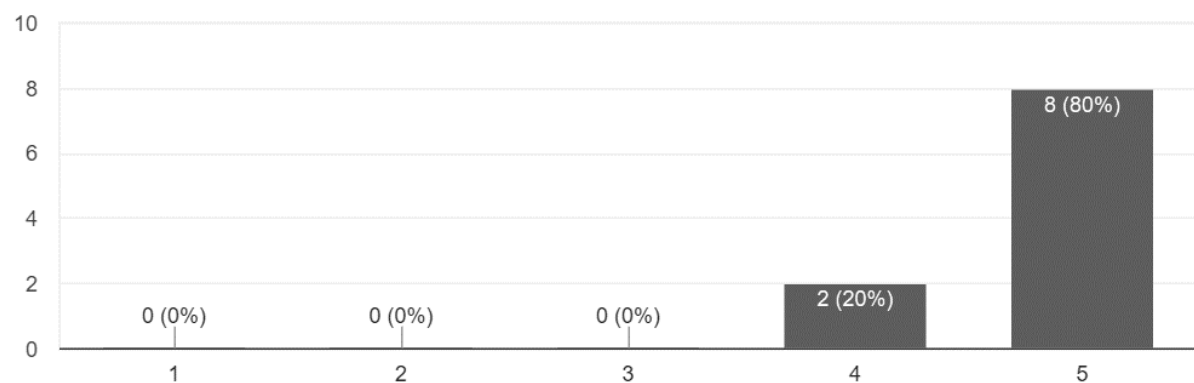


Figura 29: Resultado do item “Os níveis atingidos de saúde e de cada indicador individualmente foi facilmente visualizado na arquitetura Heal Me.”

De acordo com a distribuição das respostas, 20% concordo parcialmente e 80% concordo totalmente, pode ser indício de que os gráficos dos indicadores apresentados na

arquitetura Heal Me demonstram os níveis atingidos pela saúde e por cada um de seus cinco indicadores.

O resultado obtido para “A relação entre a saúde e seus indicadores, características e métricas foi facilmente visualizada na arquitetura Heal Me.” pode ser observado na Figura 30.

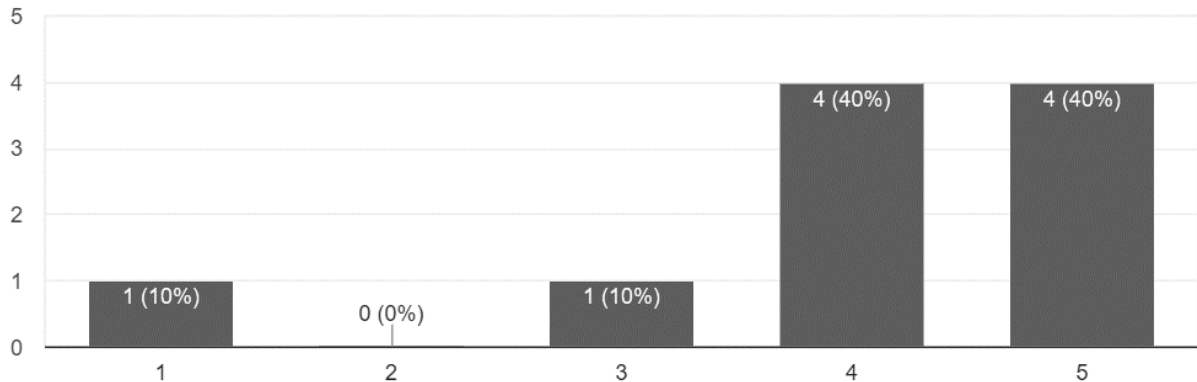


Figura 30: Resultado do item “A relação entre a saúde e seus indicadores, características e métricas foi facilmente visualizada na arquitetura Heal Me.”

A maior concentração das respostas nas opções de concordância (40% concordo parcialmente e 40% concordo totalmente) pode indicar que a relação da saúde, indicadores, características e métricas é apresentada de forma entendível para os usuários. Contudo, o fato de 10% ter avaliado como discordo totalmente e 10% ter sido indiferente pode representar a não trivialidade da composição dos processos de análise como descrito em item anterior, que será analisada nas próximas versões da arquitetura.

As respostas presentes na Figura 31 são referentes ao item “A completude de cada métrica foi facilmente visualizada na arquitetura Heal Me.”

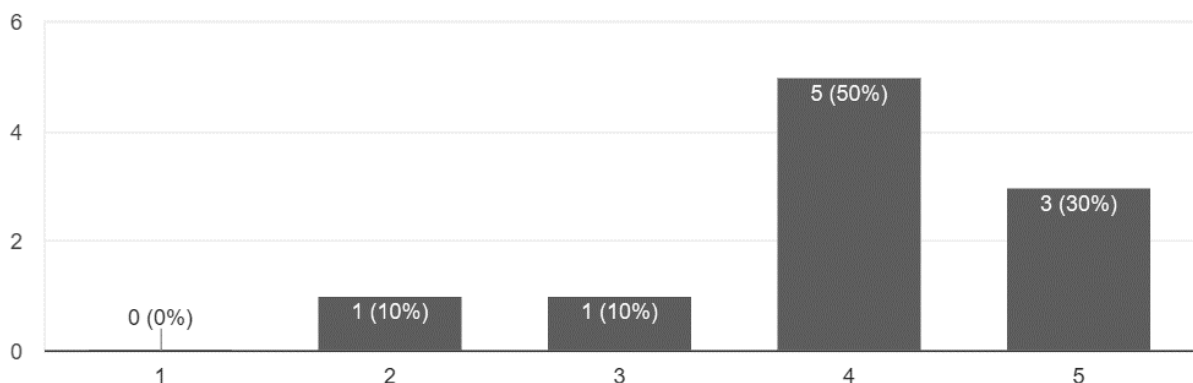


Figura 31: Resultado do item “A completude de cada métrica foi facilmente visualizada na arquitetura Heal Me.”

A completude das métricas diz respeito à sua forma de aplicação, compondo-se pelos dados capturados, os parâmetros informados, a medida, a fórmula, a unidade de interpretação

e o usuário foco da métrica. Estes dados estão presentes na arquitetura modelados através da tabela de métricas. De acordo com a concentração das respostas obtidas nas opções de concordância (50% concordo parcialmente e 30% concordo totalmente), é possível encontrar indícios de que a completude das métricas está visível na arquitetura. Contudo, a flutuação das respostas (10% de discordo parcialmente e 10% indiferente) pode indicar a necessidade de novas análises para evolução desta completude.

O quinto item diz respeito a ontologia *OntoHealth*, parte integrante da arquitetura. Esta é avaliada através do item “A composição das características e indicadores demonstra a modelagem do domínio provida pela ontologia *OntoHealth*.”, onde a resposta está presente na Figura 32.

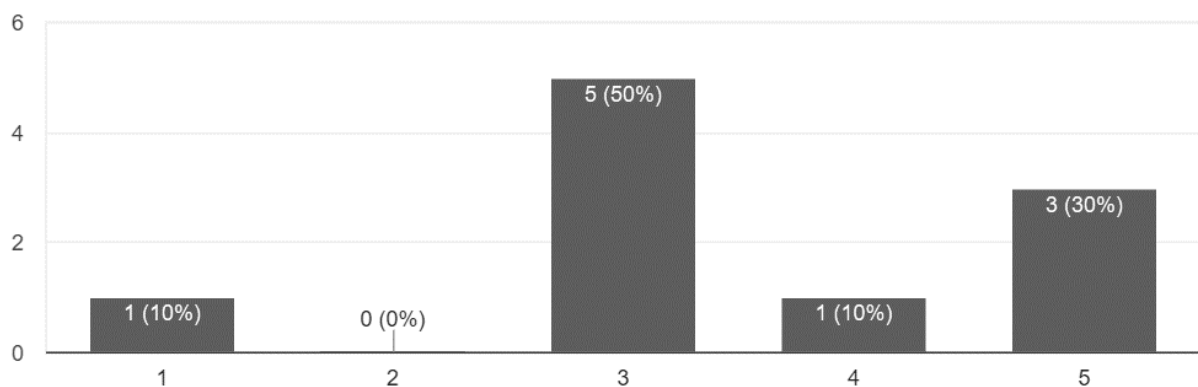


Figura 32: Resultado do item “A composição das características e indicadores demonstra a modelagem do domínio provida pela ontologia *OntoHealth*. ”

Através das respostas obtidas, pode-se observar neste caso uma dispersão dos resultados. 10% afirmaram que discordam totalmente, 50% indiferente, 10% concordam parcialmente e 30% concordam totalmente. Para prezar pela coerência, detalhes tecnológicos foram excluídos do Guia de Uso, sendo apenas mencionada a ontologia. Por este motivo, a compreensão desta pode ter sido comprometida. Este trabalho, porém, descreve minuciosamente a ontologia desenvolvida. Para o desenvolvimento de versões posteriores da documentação da arquitetura este fato será levado em consideração.

O sexto item visa avaliar os gráficos apresentados pela arquitetura como relatório das análises. Este item foi avaliado através da opção “Os gráficos visualizados na arquitetura Heal Me são de fácil compreensão.”, onde os resultados estão visíveis na figura 33.

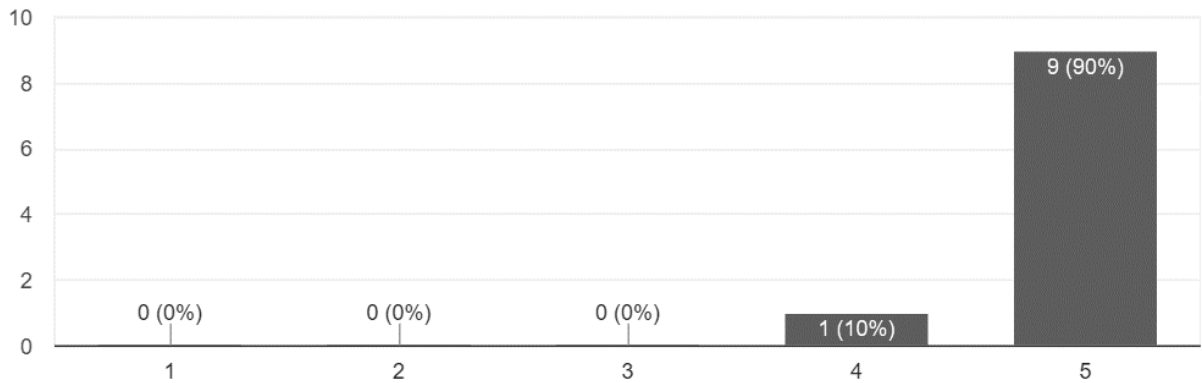


Figura 33: Resultado do item “Os gráficos visualizados na arquitetura Heal Me são de fácil compreensão”

Devido à concentração das respostas, 10% concordam parcialmente e 90% concordam totalmente, é possível identificar indícios da fácil compreensão dos gráficos apresentados no relatório das análises. Os gráficos apresentam a saúde, os indicadores individualmente, as características e o grau de atendimento de cada métrica. Foram desenvolvidos com características dinâmicas, apresentando legendas explicativas e *hints*, para facilitar a compreensão. Em acréscimo, as cores dos gráficos viabilizam a diferenciação de cada componente, melhorando a visualização.

O item que segue ainda tem foco nos gráficos da arquitetura, especificamente na identificação da saúde do ECOS utilizando a opção “Através da visualização gráfica é possível identificar corretamente a situação da saúde do ECOS avaliado.” Os resultados são visíveis na figura 34.

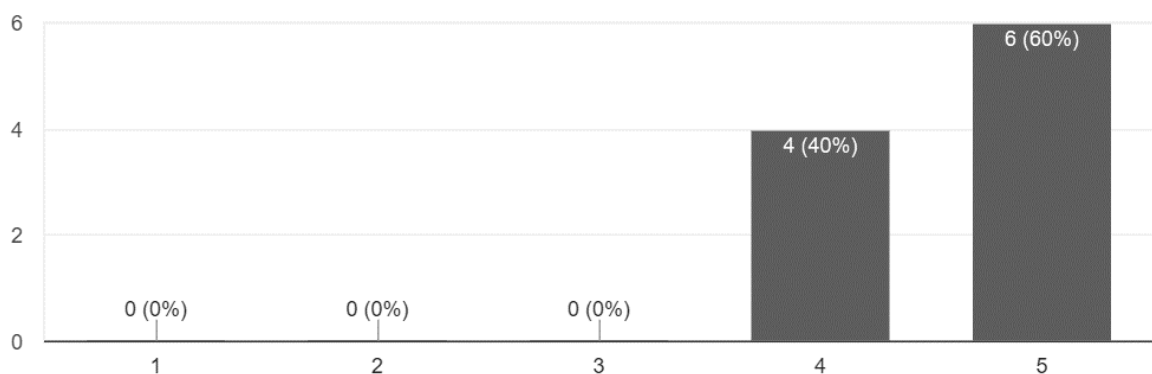


Figura 34: Resultado do item “Através da visualização gráfica é possível identificar corretamente a situação da saúde do ECOS avaliado.”

Os gráficos da arquitetura Heal Me são desenvolvidos para transformar as avaliações da saúde efetuadas pela arquitetura em escalas de porcentagem. Por adequação, são utilizados gráficos de barras para demonstrar o atendimento desta escala. Através deste cenário, os resultados de 40% concordo parcialmente e 60% concordo totalmente é possível afirmar que é

fácil a visualização do panorama de saúde através destes gráficos. Índícios desta situação são apresentados pelos resultados deste item, onde as respostas estão concentradas nos itens de maior concordância.

O oitavo item tem como foco avaliar as métricas utilizadas na arquitetura para avaliação de saúde. O item visa avaliá-las através da afirmação “As métricas presentes na arquitetura Heal Me são minimamente suficientes para a avaliação de saúde de ECOS.” As respostas encontram-se na Figura 35.

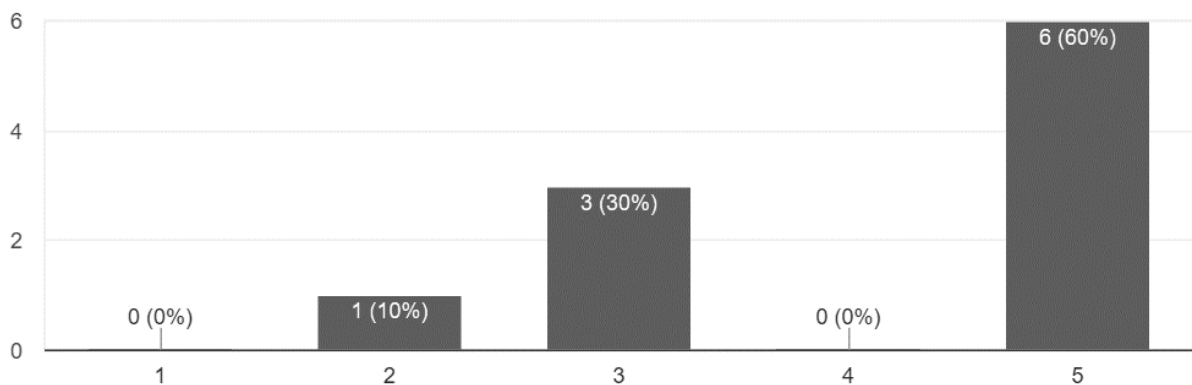


Figura 35: Resultado do item “As métricas presentes na arquitetura Heal Me são minimamente suficientes para a avaliação de saúde de ECOS”.

Apesar da concentração do maior número das respostas (60%) estar no item de maior concordância, um número significativo de respostas não atingiu níveis satisfatório (10% discordo parcialmente e 30% indiferente). Tal situação pode significar que alguns especialistas não consideram o conjunto de métrica completo para a avaliação. Através do estudo observacional apresentado, 58 métricas propostas foram avaliadas e 11 consideradas irrelevantes foram removidas. Contudo, existem diversas métricas presentes na literatura com a finalidade de avaliar a saúde de ECOS (Amorim et al. 2017). As métricas presentes na arquitetura foram um recorte inicial, gerado através dos artigos resultantes do mapeamento sistemático. Em versões posteriores da arquitetura será proposta a reavaliação das métricas, com foco em agregação por similaridade e harmonização destas métricas, para aumentar sua eficácia.

Ainda no quesito envolvendo as métricas, foi avaliada a tabela de métricas utilizadas pela arquitetura através do item “A tabela de métricas apresentada pela arquitetura Heal Me é de fácil compreensão e auxilia na utilização desta”. A Figura 36 demonstra as respostas.

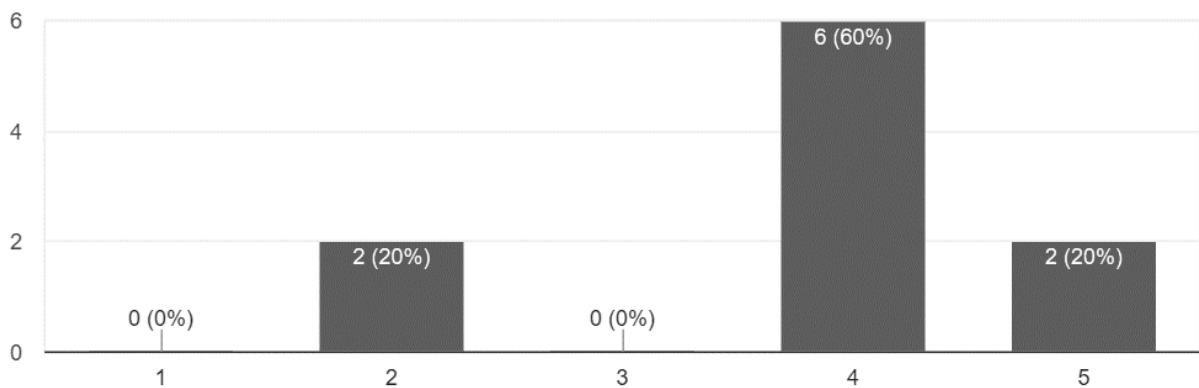


Figura 36: Resultado do item “A tabela de métricas apresentada pela arquitetura Heal Me é de fácil compreensão e auxilia na utilização desta”.

A concentração das respostas (60% concordo parcialmente e 20% concordo totalmente) apresenta indícios do atendimento deste item, indicando a possibilidade da utilidade da tabela de métricas durante a utilização da arquitetura. Contudo, a evolução das métricas, e conseqüentemente, da tabela, serão analisadas nas versões futuras com base nas notas apresentadas e nos itens anteriores já descritos.

Seguindo no quesito documentação, o item “O guia de uso apresentado torna simples a utilização da arquitetura Heal Me” teve o intuito de avaliar o guia de uso disponibilizado para a utilização da arquitetura. As respostas estão na Figura 37.

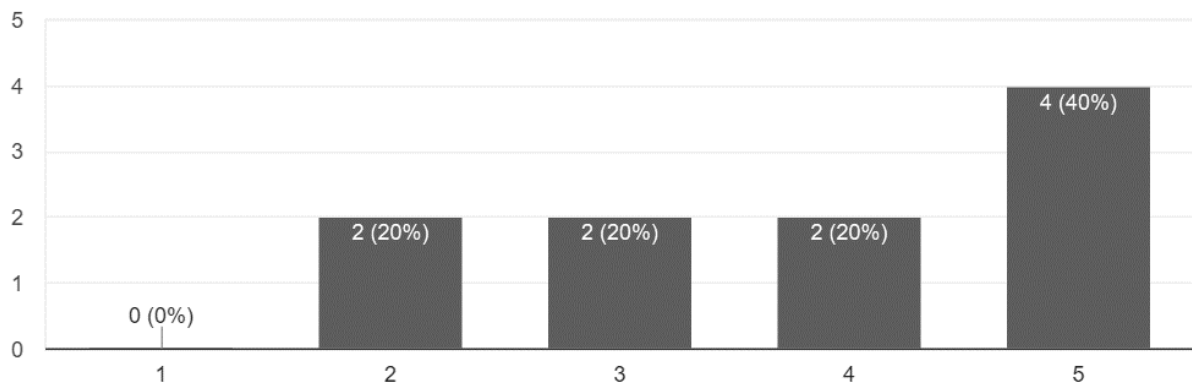


Figura 37: Resultado do item “O guia de uso apresentado torna simples a utilização da arquitetura Heal Me”.

Neste item também se encontra uma flutuação das respostas (20% discordo parcialmente, 20% indiferente, 20% concordo parcialmente e 40% concordo totalmente). O guia de uso apresenta uma descrição minuciosa dos componentes de cada tela da arquitetura, informações sobre cadastro e captura dos dados, cadastro dos parâmetros, interpretação dos gráficos das análises e utilização geral da arquitetura. Pode-se considerar que talvez um dos pontos fracos seja a grande extensão deste guia. Em versões futuras também será avaliada a objetividade e clareza do guia para agregar mais objetividade na utilização da plataforma.

Os últimos cinco itens visam uma avaliação mais genérica e de utilização da arquitetura Heal Me. O primeiro destes itens, “A arquitetura Heal Me é de fácil utilização e manuseio”, analisa a facilidade de manuseio da arquitetura e suas respostas estão presentes na Figura 38.

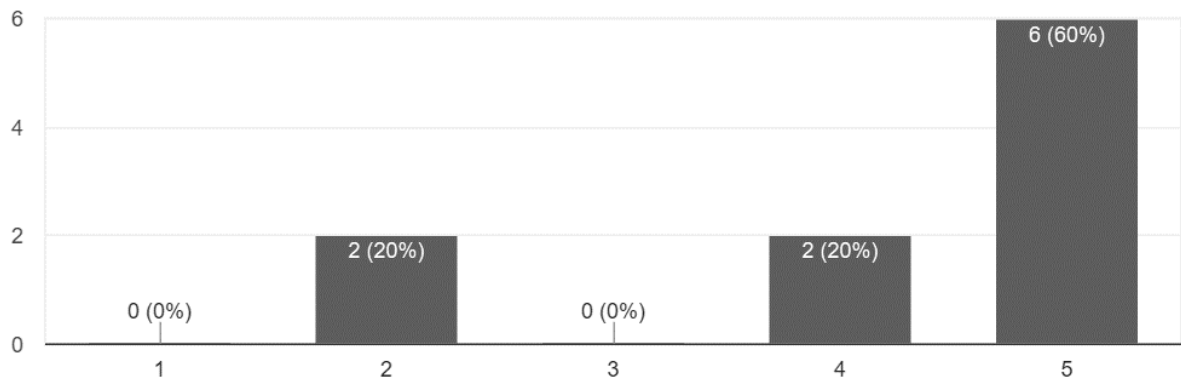


Figura 38: Resultado do item “A arquitetura Heal Me é de fácil utilização e manuseio”.

A concentração das respostas (20% concordo parcialmente e 60% concordo totalmente) apresenta indícios de que a arquitetura Heal Me pode ser manuseável. Contudo, não é possível afirmar se na visão dos participantes conhecimentos específicos são necessários para tal, além de conhecimento do domínio de ECOS. Em acréscimo, por ser um software *web*, a experiência de uso da arquitetura pode sofrer interferência da máquina utilizada para acessá-la, do *link* de internet, de servidores compartilhados, navegadores e outros componentes alheios a esta.

Para avaliar o fluxo de informações da arquitetura, foi apresentado o item “A arquitetura Heal Me é estável, não apresentando perda de informação?” onde as respostas estão presentes na Figura 39.

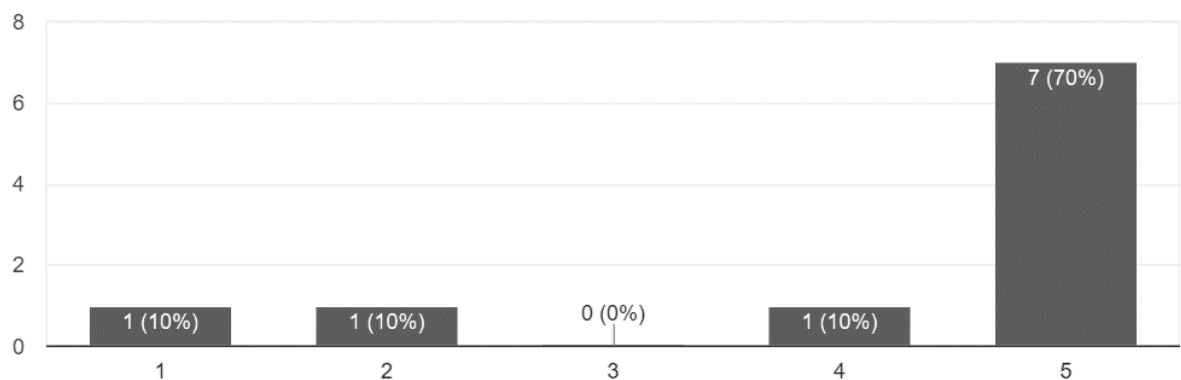


Figura 39: Resultado do item “A arquitetura Heal Me é estável, não apresentando perda de informação”.

Semelhante ao item anterior, há uma concentração das respostas em maior concordância (10% concordo parcialmente e 70% concordo totalmente), porém existem

participantes discordantes (10% discordam totalmente e 10% discordam parcialmente). Todos os comportamentos atípicos serão analisados e adequados nas próximas versões.

O último item com avaliação em escala trata do bom funcionamento da arquitetura, sendo este “A arquitetura Heal Me possui poucas ou nenhuma falha” onde as respostas estão na figura 40.

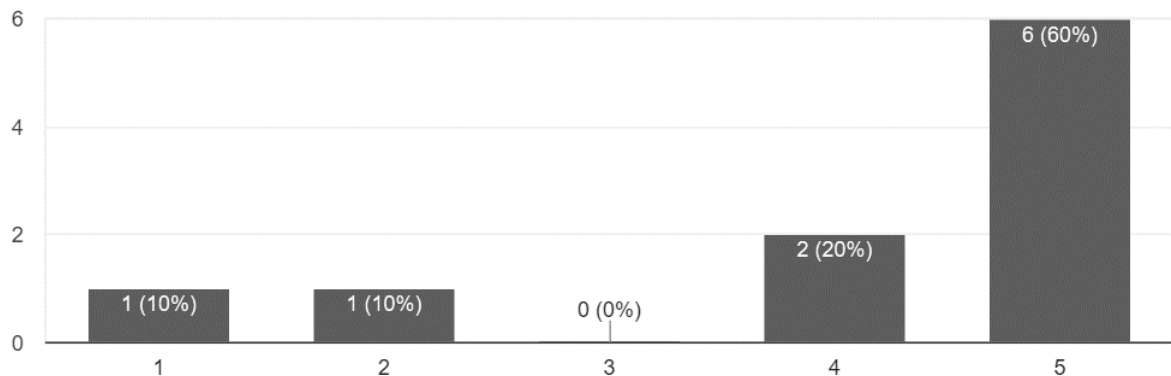


Figura 40: Resultado do item “A arquitetura Heal Me possui poucas ou nenhuma falha.”

Através do agrupamento das respostas, observa-se indícios de que o funcionamento da arquitetura como um todo tende para o adequado (20% concordo parcialmente e 60% concordo totalmente). Reitera-se, porém, que as características de software *web* podem acarretar comportamentos atípicos conforme já descrito anteriormente. Todas as possíveis situações atípicas presentes serão analisadas através de testes e solucionadas nas próximas versões.

Os três últimos itens foram planejados para capturarem as impressões de uso da arquitetura de forma descritiva. O primeiro deles, “Indique pontos positivos da arquitetura Heal Me”, pôde identificar vários aspectos da arquitetura que atraíram os especialistas. Dentre estes os mais recorrentes foram a automatização fornecida pela arquitetura, a completude das métricas que são capazes de avaliar o ambiente do ECOS de forma precisa e a possibilidade de compreensão dos resultados apresentados nas análises.

No item “Indique pontos negativos da arquitetura Heal Me”, os itens mais recorrentes foram a ausência de informações sobre a ontologia na documentação e a extensão do Guia de Uso. Estas situações já foram descritas no decorrer deste texto e serão tratadas assim como as demais críticas recebidas.

Por fim o item “Sugestões de melhoria na arquitetura Heal Me” trouxe opiniões de melhorias para as telas de cadastro, dividindo-as em etapas para facilitar o preenchimento.

Este é um recurso paliativo, uma vez que encontrar todos os dados necessários disponíveis através de APIs para o cadastro automático ainda não é possível. Outra sugestão, com base na documentação, foi a criação de um vídeo de exemplo, demonstrando a utilização da arquitetura. A criação do vídeo será realizada e este será disponibilizado junto à documentação da plataforma.

Com base nas respostas obtidas, pode-se responder as questões de pesquisa propostas. A questão “A arquitetura Heal Me funciona corretamente?” é respondida visualizando os primeiros itens do questionário. Foram observadas flutuações nestes itens, fato este que levantou suspeitas sobre comportamentos atípicos durante a utilização. Contudo, a concentração das respostas nos itens de concordância levanta indícios de que é possível que seu funcionamento tenda ao esperado. Todavia, com base nas notas gerais, é importante adotar uma estratégia de analisar o ambiente e o funcionamento da arquitetura para localizar e sanar qualquer comportamento dissonante.

Para responder à questão de pesquisa “As avaliações de saúde efetuadas pela arquitetura Heal Me são eficazes? ”, ainda utilizando as respostas obtidas como base, pode-se focar nos resultados da análises avaliadas. Através de gráficos de barras de fácil entendimento, expressando a composição dos indicadores, características e métricas, utilizando escalas de porcentagem, legendas e colocações diferentes, é possível encontrar indícios da eficiência das análises de saúde efetuadas pela arquitetura, assim como de sua compreensão. Entretanto, como o número de participantes foi reduzido, estas características da arquitetura serão novamente avaliadas, juntamente com todas as sugestões e necessidades identificadas.

4.4. AMEAÇAS À VALIDADE

O estudo de caso aqui apresentado teve por objetivo avaliar a arquitetura Heal Me e suas características como software para análise de saúde no contexto de ECOS. Entretanto, este é o primeiro esforço de implementação da arquitetura, o que torna este estudo uma avaliação preliminar. Em acréscimo, o estudo foi conduzindo com poucos especialistas. É possível também que os especialistas avaliadores não possuam profundo conhecimento em relação aos ECOS disponibilizados no estudo, apesar de terem grande conhecimento de ECOS de uma maneira geral. Por fim, o curto período de duração da avaliação, tendo durado apenas doze

dias, pode também comprometer a precisão das respostas e não permitiu a execução de um estudo piloto para calibrar o questionário aplicado e refinar o perfil dos participantes.

4.5. CONSIDERAÇÕES FINAIS DO CAPÍTULO

Este capítulo apresentou o estudo de caso executado para avaliação da arquitetura Heal Me. Foi apresentado o planejamento do estudo conduzido, incluindo o seu foco, as questões de pesquisa a serem respondidas, assim como as etapas de sua execução. Por fim, foi apresentado o perfil dos indivíduos participantes através da sua caracterização, e os resultados obtidos. Mesmo obtendo indícios mínimos do atendimento das questões de pesquisa através do agrupamento das respostas nos itens de concordância, as dispersões indicam vulnerabilidades tanto na arquitetura quanto no procedimento do estudo em si. Com base neste fato, foram apresentadas as ameaças à validade, para buscar mitigar estas vulnerabilidades e descrever o caminho de evolução que será traçado.

Todo este cenário demonstra a necessidade de novas avaliações, para redução das ameaças apresentadas e com melhorias na precisão das respostas. Será planejada a execução deste estudo em um evento especializado da área, onde o número de especialistas será maior. Agregado a isto, serão feitas as melhorias e correções já apresentadas tanto na estrutura e funcionamento da arquitetura, quando em sua documentação. O guia de uso será reformulado para atender com maior fidelidade o seu funcionamento, e será produzido um vídeo exemplificando a utilização desta.

5. CONSIDERAÇÕES FINAIS

A pesquisa contida nesta dissertação teve como objetivo apresentar a arquitetura Heal Me. A Heal Metem como objetivo efetuar a análise e avaliação semiautomática de saúde no contexto de ECOS. Através de um mapeamento sistemático foi possível identificar esta carência na literatura atual. Com base neste fato, e em outros resultados obtidos durante o mapeamento, foi possível propor uma arquitetura multicamadas, que utiliza micro serviços como unidades funcionais. Uma das principais características da arquitetura Heal Me é possuir uma ontologia como base, dotada de métricas automatizadas através de regras semânticas. Para a criação destas regras, foi efetuado um estudo observacional para avaliar as métricas e definir quais seriam utilizadas. Em sequência a este estudo, as métricas foram formalizadas e automatizadas.

Com a arquitetura desenvolvida, foi possível efetuar um estudo de caso para avaliar a utilidade e eficiência da arquitetura Heal Me em sua atividade fim. Os resultados obtidos demonstram indícios da possibilidade de sua eficácia e da fidelidade dos resultados da análise com o real panorama de saúde apresentado pelos ECOS avaliados. Foi possível encontrar nas respostas do estudo indícios das respostas das questões de pesquisa propostas por ele, derivadas da questão principal de pesquisa deste trabalho, onde: para a QP1 há a possibilidade de a arquitetura Heal Me funcionar adequadamente; e para a QP2 existem indícios de que suas avaliações de saúde podem ser eficientes. Estes indícios são visualizados na concentração das respostas dos itens do estudo nas opções de maior concordância. Entretanto, devemos ressaltar que este estudo é preliminar, apresentando pontos de melhoria e evolução, demonstrando a necessidade de novos estudos para atenuar as ameaças encontradas.

Com todo este processo, cumpriu-se o objetivo da pesquisa, onde foi proposta, apresentada e desenvolvida a arquitetura Heal Me. Os resultados obtidos, incluindo a resposta das questões QP1 e QP2, compõem a resposta da questão de pesquisa principal levantada, a saber: *Uma arquitetura de software, baseada em uma ontologia de domínio e métricas é capaz de auxiliar e automatizar a análise e a definição de saúde de Ecossistemas de Software?* Os resultados apontaram que há indícios da possibilidade de a arquitetura apresentada ser capaz de auxiliar no processo de análise de saúde de ECOS. Ainda, os resultados apontaram indícios da possibilidade de a automação deste processo contribuir com seu funcionamento.

5.1. CONTRIBUIÇÕES

Pode-se descrever algumas contribuições geradas pela abordagem apresentada nesta dissertação. A primeira contribuição é a *OntoHealth*, uma ontologia de domínio para ECOS com foco na avaliação de saúde. A modelagem oferecida pela *OntoHealth* permite uma análise profunda de todos os componentes do ECOS como sua comunidade, sua rede de relacionamento, empresas e produtos vinculados, dentre outros. Esta análise é capaz de gerar conhecimento implícito, situação a qual poderia não ser tão eficaz caso fossem usadas outras técnicas de análise de dados.

Outra contribuição importante são as regras semânticas presentes na ontologia. Estas regras visam automatizar o processo de análise de saúde, onde a aplicação das métricas da arquitetura é feita de forma automática. Esta situação também pode ter a capacidade de trazer agilidade ao processo de avaliação.

A formalização das métricas efetuada durante o desenvolvimento da arquitetura é mais uma contribuição a ser destacada. As métricas presentes na literatura e identificadas pelo mapeamento sistemático não estavam formalizadas na sua totalidade. Por estes motivos, para automatização dos processos, foi necessária esta formalização. Através desta foi possível também definir melhor o escopo de aplicação das métricas, assim como a dimensão de ECOS a qual cada uma estava vinculada. Além da automatização, este cenário pode ter a capacidade de trazer melhor precisão à aplicação das métricas e seus resultados.

O processo de análise de saúde criado para a arquitetura traz maior formalismo à atividade de análise de saúde. Fato este que entra em consonância com a ontologia, as regras semânticas e as métricas formalizadas. Este panorama formal tende a trazer maior agilidade ao processo, assim como maior precisão nos resultados.

A literatura aponta a complexidade na análise de saúde em ECOS. Devido a esta característica ser inerente ao ambiente de ECOS, os processos de análise tornam-se dispendiosos e lentos. Quando este processo é efetuado de forma manual, a situação ainda se agrava. Com base neste cenário, a semiautomatização trazida pela arquitetura Heal Me é mais uma contribuição a ser citada. Esta acontece em dois momentos: automatização da análise e obtenção dos resultados, e automatização da coleta de dados. Esta última é efetuada através do consumo de APIs vinculadas aos repositórios dos ECOS. Os serviços de captura de dados integram a plataforma de ECOS através destas APIs, capturando os dados necessários para as análises de forma automática. A agilidade causada por estes automatismos é uma das maiores

contribuições oferecidas pela arquitetura Heal Me, uma vez que a literatura tem carência destes processos.

Por fim, as análises efetuadas pela arquitetura Heal Me são de cunho genérico. O grupo de métricas foi definido buscando permitir a análise de ECOS de tipos distintos. Esta é mais uma contribuição ofertada pela arquitetura, uma vez que o ECOS a ser analisado não necessita possuir nenhuma característica específica para sua análise de saúde.

Como resultado da pesquisa desenvolvida, pode-se citar três publicações:

- Carvalho et al. 2017a: onde é apresentada a proposta inicial da arquitetura Heal Me e o grupo primitivo de métricas. Além disso, é feita a instanciação da arquitetura aplicando-a em um ECOS científico;
- Carvalho et al. 2017b: onde são discutidos conceitos de e-Learning vinculados à saúde de ECOS, e apresentada a proposta de instanciação da arquitetura em um ECOS deste tipo;
- Carvalho et al. 2018: onde é apresentado o mapeamento sistemático, o estudo observacional, as métricas formalizadas e o processo de análise de saúde proposto pela arquitetura Heal Me.

5.2. LIMITAÇÕES

A primeira limitação que pode ser descrita para a arquitetura Heal Me é a necessidade de inclusão de novas métricas, apontada de maneira indireta pela avaliação conduzida. Apesar de o processo de inclusão de novas métricas ser possível, a ontologia OntoHealth, bem como a interface web de apresentação dos resultados devem ser adequadas.

Ainda com relação às métricas, o número de participantes do estudo observacional pode ser considerado baixo. Por este motivo, estudos mais específicos para avaliar as métricas utilizadas, assim como outros grupos de métricas podem ser necessários.

Mesmo a automatização da captura dos dados sendo uma das maiores contribuições da arquitetura, infelizmente poucos são os ECOS que disponibilizam APIs com a maior parte ou todos os dados permitindo esta captura. Para atenuar esta limitação, esta versão da arquitetura Heal Me possui uma interface de cadastro manual de dados. Desta forma, o potencial de análise da arquitetura se mantém, uma vez que os dados não encontrados de forma automática podem ser inseridos e a análise assim não será comprometida.

Como foi projetada como um software web, a arquitetura Heal Me sofre efeitos de infraestrutura. Compartilhamento de servidores, disponibilidade de link de internet, compatibilidade com navegadores são alguns efeitos colaterais que podem ser encontrados durante a utilização da arquitetura. Para atenuar estes riscos, a arquitetura Heal Me foi projetada com bibliotecas cuja compatibilidade é grande com a maior parte dos navegadores. Em acréscimo, esta foi implantada em servidores cuja principal finalidade é servir softwares de cunho científico.

5.3. TRABALHOS FUTUROS

Como trabalhos futuros, pode-se descrever primeiramente o desenvolvimento de análises específicas para cada área de ECOS. A proposta é analisar e identificar as métricas de forma a definir quais métricas devem ser aplicadas em quais tipos de ECOS. ECOS científicos, de e-Learning e de negócios são alguns tipos de ECOS dos quais pode-se analisar a aplicação de métricas específicas, dentro das suas características.

Outra possibilidade é a inclusão de novas métricas. Conforme citado, atualmente a arquitetura Heal Me possui um conjunto inicial de métricas. A inclusão de outras métricas na arquitetura por parte dos usuários e seleção das métricas a serem utilizadas são funcionalidades que serão analisadas e desenvolvidas nas próximas versões da arquitetura. A análise de similaridade das métricas e harmonização, outra proposta considerada, podem permitir que o grupo de métricas tenha maior eficiência e eficácia para o processo de avaliação de saúde.

Com a evolução da arquitetura, será possível criar um modelo de dados padrão para a captura. Este modelo poderá ser adotado pelas APIs de repositório, o que viabilizará a captura automática de dados de forma mais precisa. Em acréscimo, a evolução da camada de conversão, tratando um maior número de modelos de dados diferentes, também viabilizará esta atividade.

A proposta de evoluir a arquitetura para uma ideia colaborativa também é uma atividade futura. Permitir que os dados sejam informados por diversos usuários, de forma colaborativa e ao longo do tempo, pode permitir uma avaliação controlada com o passar do tempo. Em acréscimo, é possível que um maior número de dados seja capturado, permitindo maior precisão às análises.

Por fim, a arquitetura poderá dispor de um sistema de níveis de saúde. Através dos resultados obtidos durante as análises, as plataformas poderão ser classificadas de acordo com os níveis de saúde obtidos, níveis de seus indicadores e características. Esta classificação permitirá fidelizar ainda melhor os resultados obtidos, além de permitir uma melhor comparação histórica das análises.

REFERÊNCIAS

- AUDY, J. L. N. e PRIKLADNICKI, R. **Desenvolvimento Distribuído de Software**. Rio de Janeiro: Elsevier, 2007.
- ALVES, A. M. e PESSOA, M. Brazilian public software: beyond sharing. **In Proceedings of the International Conference on Management of Emergent Digital EcoSystems**, pages 73–80. ACM, 2010.
- ALVES, A. M. PESSOA, M. e SALVIANO, C. F. Proposal for a framework for quality measurement to the spb–brazilian public software. **Business Process Management Journal**, 21(1):100–125, 2015.
- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS - ABNT. **NBR ISO/IEC 9126-1:2033** - Engenharia de software - Qualidade de Produto. Parte1: Modelo de Qualidade. Rio de Janeiro, 2003
- AXELSSON, J. &SKOGLUND, M. Quality assurance in software ecosystems: A systematic literature mapping and research agenda. **Journal of Systems and Software**, 114, 69-81. 2016.
- BARBOSA, O. SANTOS, R.P. ALVES, C. WERNER, C. e JANSEN, S. A systematic mapping study on software ecosystems from a three-dimensional perspective. **Software Ecosystems: analyzing and managing business networks in the Software Industry**. 2013. p. 59-81.
- BASILI, V. CALDEIRA, G. ROMBACH, H. Goal Question Metric Approach, **Encyclopedia of Software Engineering**, John Wiley & Sons, Inc., p. 528-532, 1994.
- BEDOYA, F. HERNÁN, Ó. AMELLER, D. COSTA, D. &GUTIÉRREZ, J. **QuESo V2.0**: a quality model for open source software ecosystems: List of measures. 2016. (Technical Report).
- BERGER, T. PFEIFFER, R.-H. TARTLER, R. DIENST, S. CZARNECKI, K. WSOWSKI, A. SHE, S. Variability mechanisms in software ecosystems. **Inf. Softw. Technol.** 56 (11), 1520–1535. 2014.
- BOSCH, J. From software product lines to software ecosystems. **In Proceedings of the 13th International Software Product Line Conference**, SPLC '09, pages 111–119, Pittsburgh, PA, USA, 2009. Carnegie Mellon University.
- BOSCH, J. 2016. **Speed, Data, and Ecosystems**: Excelling in a Software-Driven World. CRC Press, Inc., Boca Raton, FL, USA.
- BOSCH, J. e BOSCH-SIJTSEMA, P. M. Softwares product lines, global development and ecosystems: collaboration in software engineering. **In Collaborative Software Engineering**, pages 77–92. Springer, 2010
- BPMN, Object Management Group Business Process Model and Notation. 2018. Disponível em < <http://www.bpmn.org/> >. Acessado em 26 de agosto de 2018.

- BUSCHMANN, F. HENNEY, K. SCHMIDT, D. C. **Pattern-Oriented Software Architecture**, Volume 4: A Pattern Language for Distributed Computing. Chichester, UK: Wiley. 2007.
- CARVALHO, I. CAMPOS, F. BRAGA, R. DAVID, J. M. STROELE, V. ARAUJO, M. A. **HEAL ME - An Architecture for Health Software Ecosystem Evaluation. In Proceedings of the 5th International Workshop on software Engineering for Systems-of-Systems and 11th Workshop on Distributed Software Development, Software Ecosystems and Systems-of-System.** In ICSE. IEEE. 2017a.
- CARVALHO, I. VEIGA, W. CAMPOS, F. C. A. BRAGA, R. M. STRÖELE, V. **Qualidade de um Ecosistema de e-Learning: Indicadores de Saúde.** In: Simpósio Brasileiro de Sistemas de Informação, 2017, Lavras. **Anais.** 13ª Simpósio Brasileiro de Sistemas de Informação (SBSI2017), 2017b.
- CARVALHO, I. CAMPOS, F. BRAGA, R. DAVID, J. M. N. STROELLE, V. ARAUJO, M. A. **Health Evaluation in Software Ecosystems.** In: 20th International Conference on Enterprise Information Systems (ICEIS 2018), Santa Cruz - Madeira. **Proceedings of the 20th International Conference on Enterprise Information Systems (ICEIS 2018).** Lisboa: SCITEPRESS, 2018. v. 2. p. 263-271.
- CLEMENTS, P. NORTHROP, L. **Software Product Lines: Practices and Patterns.** Boston: Addison-Wesley, 2002, 563 p.
- D. DHUNGANA, I. GROHER, E. SCHLUDERMANN, E S. BIFFL. **Software ecosystems vs. natural ecosystems: Learning from the ingenious mind of nature. In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume, ECSA '10,** pages 96–102, New York, NY, USA. ACM. 2010.
- ECLIPSE. **Eclipse.org.** Disponível em <<http://www.eclipse.org/>>. Acessado em 03 de setembro de 2018.
- ELMASRI, R. NAVATHE, S. B. & PINHEIRO, M. G. **Sistemas de banco de dados.** São Paulo: Pearson. 2005.
- FRANCO-BEDOYA, O. AMELLER, D. COSTAL, D. e FRANCH, X. **QuESo a quality model for open source software ecosystems. In Software Engineering and Applications (ICSOFT-EA), 2014 9th International Conference on,** pages 209–221, Aug 2014.
- FRANZ, R. Z., SAWICKI, S., ROOS-FRANTZ, F., YEVSEYEVA, I., & EMMERICH, M. T. **On using Markov decision processes to model integration solutions for disparate resources in software ecosystems.** 2015. Springer.
- FREITAS, V. et al. **Uma arquitetura para Ecosistema de Software Científico. Workshop em Desenvolvimento Distribuído de Software, Ecosistemas de Software e Sistemas-de-Sistemas,** p. 41-48. 2015.
- GIMENES, I. e TRAVASSOS, G. H. **O Enfoque de Linha de Produto para Desenvolvimento de Software.** XXI Jornada de Atualização em Informática - Livro Texto, Porto Alegre. 2002.
- GITHUBAPI. **GitHub.** Disponível em <<https://developer.github.com/v3/>>. Acessado 27 de agosto 2018.

- HIBERNATE. **Hibernate**. Disponível em <<http://hibernate.org/>>. Acessado em 26 de agosto de 2018.
- HMOOD, A. KEIVANLOO, I. E RILLING, J. Se-Equam - an evolvable quality metamodel. **In Computer Software and Applications Conference Workshops (COMPSACW)**, 2012 IEEE 36th Annual, pages 334–339, July 2012.
- HORROCKS, I. PATEL-SCHNEIDER, P. F. BOLEY, H. TABET, S. GROSOFF, B. e DEAN, M. **SWRL: A semantic web rule language combining OWL and Rule ML**. W3C Member submission, 21, 79. 2004.
- JANSEN, S. How quality attributes of software platform architectures influence software ecosystems. **In Proceedings of the 2013 International Workshop on Ecosystem Architectures**, WEA 2013, pages 6–10, New York, NY, USA, 2013. ACM.
- JANSEN, S. FINKELSTEIN, A. e BRINKKEMPER, S. A sense of community: A research agenda for software ecosystems. **In Software Engineering-Companion Volume**, 2009. ICSE-Companion 2009. 31st International Conference on (pp. 187-190). IEEE.
- JANSEN, S., M. A. CUSUMANO, e BRINKKEMPER, S. E. **Software Ecosystems: Analyzing and Managing Business Networks in the Software Industry**. Edward Elgar Publishing, 2013.
- JANSEN, S. Measuring the health of open source software ecosystems: Beyond the scope of project health. **Information and Software Technology** 56.11. 2014: 1508-1519.
- JAVA. **Oracle Jc**. Disponível em <https://www.java.com/pt_BR/about/>. Acessado 27 de agosto de 2018
- JENA. **Apache Jena**. Disponível em <<https://jena.apache.org/>>. Acessado em 27 de agosto de 2018.
- KAJAN, E. LAZIĆ, L. & MAAMAR, Z. Software engineering framework for digital service-oriented ecosystem. **In Telecommunications Forum (TELFOR)**, 2011 19th (pp. 1320-1323). IEEE.
- KDE. **KDE.org**. Disponível em <<https://www.kde.org/>>. Acessado 03 de setembro de 2018.
- KITCHENHAM, B. A. **Procedures for performing systematic reviews**. Keele, UK, Keele University, 33(2004):1–26, 2004.
- KITCHENHAM, B. A. DYBA, T. e JORGENSEN, M. Evidence-based software engineering. **In Proceedings of the 26th international conference on software engineering**, pages 273–281. IEEE Computer Society, 2004.
- LIKERT, R. **A technique for the measurement of attitudes**. Archives of psychology, 1932.
- LINGEN, S.V. PALOMBA, A. LUCASSEN, G. On the software ecosystem health of open source content management systems. In: Alves, C.F., Hanssen, G.K., Bosch, J., Jansen, S. (Eds.), **Proceedings of the 5th International Workshop on Software Ecosystems**, Potsdam, Germany, Vol. 987, pp. 45–56. June 11, 2013.

- LYTRA, I., ENGELBRECHT, G., SCHALL, D., & ZDUN, U. Reusable architectural decision models for quality-driven decision support: A case study from a smart cities software ecosystem. **In Software Engineering for Systems-of-Systems (SESoS)**, 2015 IEEE/ACM 3rd International Workshop on (pp. 37-43). IEEE.
- MHAMDIA, A. B. H. S. Performance measurement practices in software ecosystem. **International Journal of Productivity and Performance Management**, 62(5):514–533, 2013.
- MANIKAS, K. Revisiting software ecosystems research: a longitudinal literature study. 2016. **Journal of Systems and Software**, 117, 84-103.
- MANIKAS, K., & HANSEN, K. M. Reviewing the health of software ecosystems—a conceptual framework proposal. **In Proceedings of the 5th International Workshop on Software Ecosystems (IWSECO)**, pages 33-44. 2013.
- PAUTASSO, C. ZIMMERMANN, O. & LEYMANN, F. Restful web services vs. big’ web services: making the right architectural decision. **In Proceedings of the 17th international conference on World Wide Web** (pp. 805-814). 2008.ACM.
- PERRY, D.E. SIY, H.P. AND VOTTA, L. G. Parallel changes in large scale software development: an observational case study. **International Conference on Software engineering (ICSE)** (Kyoto, Japan, Apr. 1998), 251–260.
- PETTICREW, M. e ROBERTS, H. **Systematic reviews in the social sciences: A practical guide**. John Wiley & Sons, 2008.
- RUNEON, P. HOST, M. RAINER, A. REGNELL, B. **Case study research in software engineering: Guidelines and examples**, 2012.
- SANTOS, R. VALENÇA, G. VIANA, D. ESTÁCIO, B. FONTÃO, A. MARCZAK, S. WERNER, C. ALVES, C. CONTE, T. E PRIKLADNICKI, R. Qualidade em ecossistemas de software: Desafios e oportunidades de pesquisa. **In Proceedings of VIII Workshop on Distributed Software Development, Software Ecosystems and Systems-of-Systems**, pages41–44, 2014 (in portuguese).
- SANTOS, R. **Managing and Monitoring Software Ecosystem to Support Demand and Solution Analysis**. 2016. 246f. Tese (Doutorado em Engenharia de Sistemas e Computação) - Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE. Universidade Federal do Rio de Janeiro, Rio de Janeiro.
- SCHUGERL, P. RILLING, J. WITTE, R. & CHARLAND, P. A quality perspective of software evolvability using semantic analysis. **In Semantic Computing**, 2009. ICSC'09. IEEE International Conference on (pp. 420-427). IEEE.
- da SILVA AMORIM, S. et al. How Has the Health of Software Ecosystems Been Evaluated? A Systematic Review. **In: Proceedings of the 31st Brazilian Symposium on Software Engineering**. ACM, 2017. p. 14-23.
- da SILVA AMORIM, S. MCGREGOR, J. D. DE ALMEIDA, E. S.VON FLACH, G. CHAVEZ, C. Flexibility in ecosystem architectures. **In Proceedings of the 2014 European Conference on Software Architecture Workshops** (p. 14). 2014. ACM.

- da SILVA AMORIM, S. MCGREGOR, J. D. DE ALMEIDA, E. S. e CHAVEZ, C. V. F. G. Understanding the Effects of Practices on KDE Ecosystem Health. **In IFIP International Conference on Open Source Systems**. Springer, Cham. 2017. pp. 89-100.
- SIRIN, E., PARSIA, B., GRAU, B. C., KALYANPUR, A., & KATZ, Y. Pellet: A practical owl-dl reasoner. **Web Semantics: science, services and agents on the World Wide Web**, 5(2), 51-53. 2007.
- SOTO, M. e CIOLKOWSKI, M.2009. The QualOSS open source assessment model measuring the performance of open source communities. **In Proceedings of the 3rd ESEM**, pages 498–501.
- STEFANUTO, G. SPIESS, M. ALVES, A. M. & CASTRO, P. F. Quality in software digital ecosystems the user's perceptions. **In Proceedings of the International Conference on Management of Emergent Digital EcoSystems** (pp. 85-88). 2011. ACM.
- van den BERK, I. JANSEN, S. & LUIJNENBURG, L. Software ecosystems: a software ecosystem strategy assessment model. **In Proceedings of the Fourth European Conference on Software Architecture: Companion Volume** (pp. 127-134). 2010. ACM.
- WORDPRESS. **Wordpress.org**. Disponível em <<https://wordpress.org/>>. Acessado 28 de agosto de 2018.
- WU, A. **Taking the Cloud-Native Approach with Microservices**. Google ic. 2017. Disponível em <<https://cloud.google.com/files/Cloud-native-approach-with-microservices.pdf>>. Acessado 27 de agosto de 2018.
- YIN, R. K. **Estudo de Caso: Planejamento e Métodos**. Bookman editora, 2015.

APÊNDICE A - TERMO DE CONSENTIMENTO DE PARTICIPAÇÃO

Responsáveis:

- Iuri Andrade Carvalho – Pesquisador
- Fernanda Campos – Orientadora
- Regina Braga - Co orientadora

Este é um termo de consentimento para participar voluntariamente do estudo: “Uma arquitetura para avaliação de saúde de Ecossistemas de Software”. Por favor, leia com atenção as informações abaixo antes de dar seu consentimento para participar do estudo. Qualquer dúvida pode ser esclarecida diretamente com o pesquisador através do e-mail: iuri.carvalho@ice.ufjf.br.

Objetivo e Benefícios do Estudo

Este estudo tem por objetivo avaliar a arquitetura denominada Heal Me, destinada a avaliação de saúde de ecossistemas de software (ECOS).

Procedimentos

Cada avaliador irá utilizar as funcionalidades disponibilizadas pela arquitetura e avaliar sua eficácia, clareza e facilidade de uso. Requisitos de qualidade, como estabilidade e robustez também serão avaliados. Você irá utilizar a ferramenta, observar os resultados apresentados pelas análises efetuadas, e avaliará os procedimentos através de um questionário.

Participação Voluntária

A sua participação neste estudo é voluntária e terá plena e total liberdade para desistir do estudo a qualquer momento, sem que isso acarrete qualquer prejuízo.

Garantia de Sigilo e Privacidade

As informações relacionadas ao estudo são confidenciais e qualquer informação divulgada em relatório ou publicação será feita sob forma codificada, para que a confidencialidade seja mantida. O pesquisador garante que seu nome não será divulgado sob hipótese alguma.

Ao marcar sim na opção abaixo e seguir com a execução da avaliação, declaro que fui esclarecido sobre os objetivos, procedimentos e benefícios do presente estudo. Participo de livre e espontânea vontade do estudo em questão. Foi-me assegurado o direito de abandonar o estudo a qualquer momento, se eu assim o desejar. Declaro também não possuir nenhum grau de dependência profissional ou educacional com os pesquisadores envolvidos nesse projeto (ou seja, os pesquisadores desse projeto não podem me prejudicar de modo algum no trabalho ou nos estudos), não me sentindo pressionado de nenhum modo a participar dessa pesquisa.

Sim Não

APÊNDICE B - FORMULÁRIO DE CARACTERIZAÇÃO DO PARTICIPANTE

Nome: _____

1) Formação Acadêmica: Pós-Doutorado Doutorado Mestrado Especialização
Graduação Técnico Outra: _____

2) Atualmente trabalha na: Academia Indústria Academia e Industria

Tempo na academia (em anos): _____

Tempo da indústria (em anos): _____

3) Tem conhecimento/experiência com Ecossistemas de Software (ECOS)? Não Sim

- Tempo (em anos): _____

4) Já trabalho em ambiente de ECOS? Não Sim

- Tempo (em anos): _____

Experiência em Ecossistemas de Software

Informe o tempo de trabalho em ambiente de ECOS e indique o grau de sua experiência para cada item nesta seção, seguindo a escala de 5 pontos abaixo:

1 = nenhum

2 = conhecimento teórico (estudei em aula ou em livro ou assisti palestra/aula sobre ECOS)

3 = participei em um ou mais ambientes de ECOS na indústria ou academia, mas sem ser o responsável

4 = participei em até 03 oportunidades na indústria ou academia como responsável

5 = participei em mais de 03 oportunidades na indústria ou academia como responsável incluindo diferentes organizações e/ou diferente tipos de ECOS.

5) Experiência como desenvolvedor/colaborador em ambiente de ECOS: 1 2 3 4 5

Tempo: _____

6) Experiência como gerente/responsável por processos em ECOS: 1 2 3 4 5

Tempo: _____

7) Experiência em definir arquitetura de plataforma/ambiente em ECOS: 1 2 3 4 5

Tempo: _____

8) Experiência com avaliação/controle de qualidade em ECOS: 1 2 3 4 5

Tempo: _____

9) Experiência com avaliação/controle de saúde em ECOS: 1 2 3 4 5

Tempo: _____

APÊNDICE C - AVALIAÇÃO DA ARQUITETURA HEAL ME

1) O panorama geral de saúde do ecossistema de software (ECOS) avaliado foi facilmente compreendido na arquitetura HEAL ME.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

2) Os níveis atingidos de saúde e de cada indicador individualmente foi facilmente visualizado na arquitetura HEAL ME.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

3) A relação entre a saúde e seus indicadores, características e métricas foi facilmente visualizado na arquitetura HEAL ME.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

4) A completude de cada métrica foi facilmente visualizada na arquitetura HEAL ME.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

5) A composição das características e indicadores demonstra a modelagem do domínio provida pela ontologia OntoHealth.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

6) Os gráficos visualizados na arquitetura HEAL ME são de fácil compreensão.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

7) Através da visualização gráfica é possível identificar corretamente a situação da saúde do ECOS avaliado.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

8) As métricas presentes na arquitetura HEAL ME são minimamente suficientes para a avaliação de saúde de ECOS.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

9) A tabela de métricas apresentada pela arquitetura HEAL ME é de fácil compreensão e auxilia na utilização desta.

1. Discordo totalmente

2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

10) O guia de uso apresentado torna simples a utilização da arquitetura HEAL ME.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

11) A arquitetura HEAL ME é de fácil utilização e manuseio.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

12) A arquitetura HEAL ME é estável, não apresentando perda de informação.

1. Discordo totalmente
2. Discordo parcialmente
3. Indiferente
4. Concordo parcialmente
5. Concordo totalmente

13) A arquitetura HEAL ME é robusta, com poucas ou nenhuma falha.

6. Discordo totalmente
7. Discordo parcialmente
8. Indiferente
9. Concordo parcialmente
10. Concordo totalmente

14) O que mais gostou na arquitetura HEAL ME?

15) O que menos gostou na arquitetura HEAL ME?

16) O que mudaria na arquitetura HEAL ME?

APÊNDICE D - GUIA DE USO – ARQUITETURA HEAL ME

1. INTRODUÇÃO

A arquitetura Heal Me tem como objetivo principal avaliar a saúde do ECOS utilizando regras semânticas aplicadas aos seus. A análise é feita em dados do ECOS, seus produtos e ambiente. As métricas são aplicadas a esses dados, de acordo com cada um dos cinco indicadores de saúde, descritos na tabela de métricas disponível na arquitetura. Após o processamento destas informações, é possível avaliar as condições de saúde da plataforma ECOS, bem como suas restrições. Como apoio a este Guia de Uso a arquitetura Heal Me disponibiliza sua tabela de métricas, que pode ser acessada através do menu *Documentation*, opção *Metric Table*.

2. PROCESSO DE UTILIZAÇÃO DA ARQUITETURA HEAL ME

O funcionamento da Arquitetura Heal Me é baseado no processo de análise de saúde apresentado na Figura 1. Este processo engloba todas as etapas da análise, desde a obtenção dos dados, da percepção do usuário sobre o ambiente, até a exibição dos resultados da análise

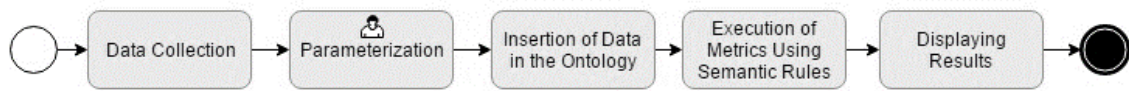


Figura 1: Processo de análise de saúde.

Passo1: O primeiro passo do processo é o lançamento dos dados do ECOS. Esta etapa é acessada na arquitetura através do menu *Registration*, opção *Platforms* como mostra a Figura 2. A Figura 3 ilustra a tela de cadastro.

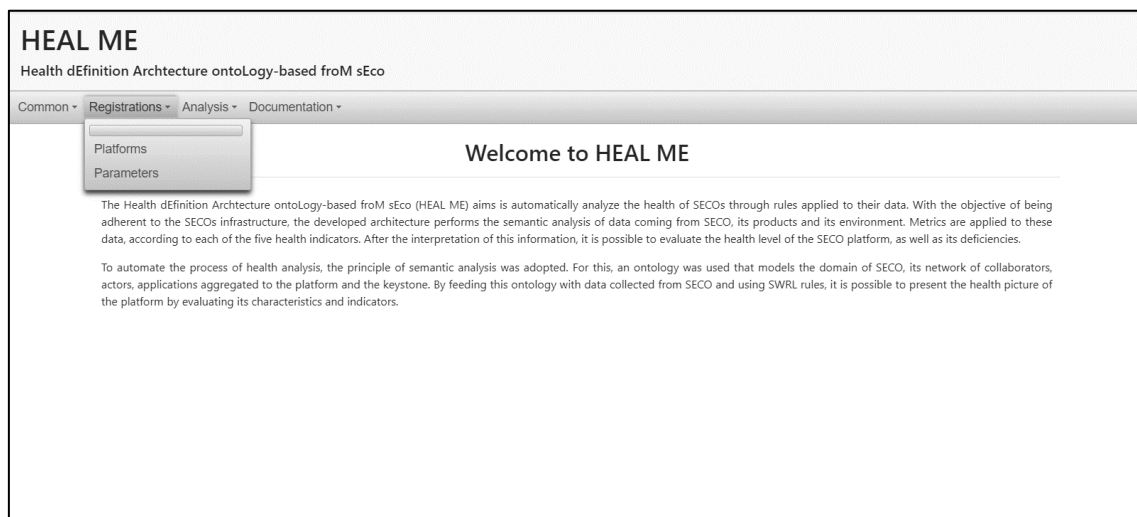


Figura 2: Acesso à página de cadastro.

HEAL ME
Health dEfinition Architecture ontoLogY-based froM sEco

Common ▾ Registrations ▾ Analysis ▾ Documentation ▾

Platform Registration

Register the platform or find the platform which you want update.

Auto Import

Select

Platform:

Name

Platform Name:

Sustainability

Heterogeneity

Number of countries: Semantic closeness: Node types:

Regeneration Ability

Working Time Set: New Members:

Effort Balance

Number of Commits: Active Members: Publications:

Figura 3: *Registration Platform.*

Passo 1.1: Esta tela tem o intuito de inserir ou editar os dados do ECOS a ser analisado. Para inserir uma nova análise, basta clicar no botão *New*, preencher os dados e clicar no botão *Save*.

Para edição dos dados já cadastrados anteriormente, basta selecionar a plataforma no campo de seleção *Platform*, e em seguida clicar no botão *Load* como demonstra a Figura 4. Após carregados, basta edita-los e clicar novamente no botão *Save*.

HEAL ME
Health dEfinition Architecture ontoLogY-based froM sEco

Common ▾ Registrations ▾ Analysis ▾ Documentation ▾

Platform Registration

Register the platform or find the platform which you want update.

Auto Import

Select

Platform:

Name

Platform Name:

Sustainability

Heterogeneity

Number of countries: Semantic closeness: Node types:

Regeneration Ability

Working Time Set: New Members:

Effort Balance

Number of Commits: Active Members: Publications:

Figura 4: Seleção de plataforma para edição.

Cada campo é anotado com o nome da métrica que representa, e deve ser preenchido com o dado referente à medição efetuada para aplicação da métrica. Cada dado representa uma métrica que será utilizada na análise. Estes podem ser informados manualmente preenchendo o campo referente à sua métrica. Para referenciar cada dado na tabela de

métricas, basta posicionar o mouse sobre a etiqueta de nome do campo e verificar no *hint* onde localiza-lo como demonstra a Figura 5.

The image shows a form field with the label "Number of countries :". The input field contains the value "0". A black tooltip box with white text is positioned below the label, containing the text "See the specification of metric 1 in Metric Table".

Figura 5: *Hint* de identificação dos campos.

Exemplo: A plataforma que se deseja analisar possui comunidades em 52 países. Para preencher este dado, que é referente à métrica 1 (número de países) da arquitetura, basta localizar o campo no formulário de registro e visualizar o *hint* com sua identificação na tabela de métricas. Após, basta acessar a tabela através do menu *Documentation*, opção *MetricTable*. Tendo visualizado as informações sobre a métrica na tabela, basta preencher o campo com o dado de número de países. Após o preenchimento dos demais dados, basta clicar no botão *Save* como já descrito.

The image shows a registration form with several sections, each containing input fields with the value "0":

- Interrelatedness**: Number of Connections, Connectivity Capacity, Ratio Between Number of Connections and Capacity, Nodes Centrality, External Partners.
- Information Consistence**: Glossary of Terms.
- Clustering**: Product Types, Greater Collaboration, Active Projects, Total Produced Files.
- Financial Consistence**: Number of Partners, Commercial Sponsorship, Total Contribution Value, Active Contributors, Frequently Users.
- Niche Creation**: Documentation, Types of Contributors, Types of Application Projects, Natural Language Support, Supported Technologies, Supported Development Technologies.

At the bottom of the form, there are three buttons: "New", "Save", and "Cancel".

Figura 6: Botões de opção do formulário.

Passo 1.2: A arquitetura Heal Me permite a captura de dados de forma automática. Esta captura funciona consumindo *APIs* distribuídas pelos repositórios dos ECOS, e extraindo destes os dados através de *endpoints*. Para efetuar a extração automática de dados, basta clicar no botão *AutoImport*, informar os *endpoints* vinculados às métricas que se deseja preencher e após clicar no botão *Import*. Esta funcionalidade é demonstrada na Figura 7.

Metric	EndPoint
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>

Figura 7: Cadastro de *endpoints* para captura automática de dados.

Exemplo: Deseja-se cadastrar o número de novos membros automaticamente. O repositório da plataforma possui o *endpoint* <http://wsteste/platformteste/numberofnewmember> que disponibiliza este dado. Para o cadastro, basta clicar no botão *AutoImport*, após selecionar a métrica *Number of New Members* no *dropdown* da opção *Metric*. Por fim, preencher o campo *EndPoint* com o *endpoint* <http://wsteste/platformteste/numberofnewmember> e clicar no botão *Import*.

Algumas APIs disponibilizam seus dados em formatos muito específicos, sendo necessário criação de objetos de domínio e apontamentos para “desserialização”. Para estes casos é necessário o desenvolvimento de uma camada de integração, para conversão dos dados e alimentação da arquitetura.

Passo 2: Após cadastrada a plataforma, deve-se informar os parâmetros da análise. Estes são necessários para a comparação dos dados cadastrados. Informar os parâmetros é a forma do usuário que irá efetuar a análise transmitir à arquitetura sua percepção sobre o ambiente que será analisado. Para cadastrar os parâmetros, basta acessar o menu *Registrations*, opção *Parameters*. Após, na caixa de seleção *Platform*, seleciona-se a plataforma a qual os parâmetros serão informados e clica-se no botão *Load*.

Figura 8: Cadastro dos parâmetros da análise

Figura 9: Opções do formulário de cadastro de parâmetros.

Cada parâmetro é referente a uma métrica e a um dado informado no cadastro. Os dados preenchidos serão comparados a estes parâmetros, para avaliação das métricas. Para visualizar a qual dado e a qual métrica o parâmetro está vinculado, basta visualizar o *hint* de ajuda na etiqueta de nome do parâmetro como demonstra a Figura 5. Após o preenchimento dos parâmetros, basta salvá-los clicando no botão *Save*.

Exemplo: Pelo porte da plataforma que se deseja avaliar, espera-se que ela esteja presente em no mínimo 50 países. Para cadastrar este parâmetro, deve-se acessar o menu Registration, opção Parameters. Na página de cadastro, deve-se selecionar a plataforma *Platform Test* no *dropdown Platforms*, e clicar no botão *Load*. Após, deve se procurar o campo referente *Number of Countries* e preenche-lo como valor 50. Após os demais parâmetros terem sido preenchidos, deve-se clicar no botão *Save* para salvar os parâmetros.

Caso haja dúvidas, basta observar o hint presente no campo e acessar a métrica referente na tabela de métricas.

Passo 3: Inseridos os dados do ambiente e da plataforma do ECOS, é possível efetuar a análise. Para isto, basta acessar a opção *Analysis* do menu, e posteriormente a opção *Health Analysis*.

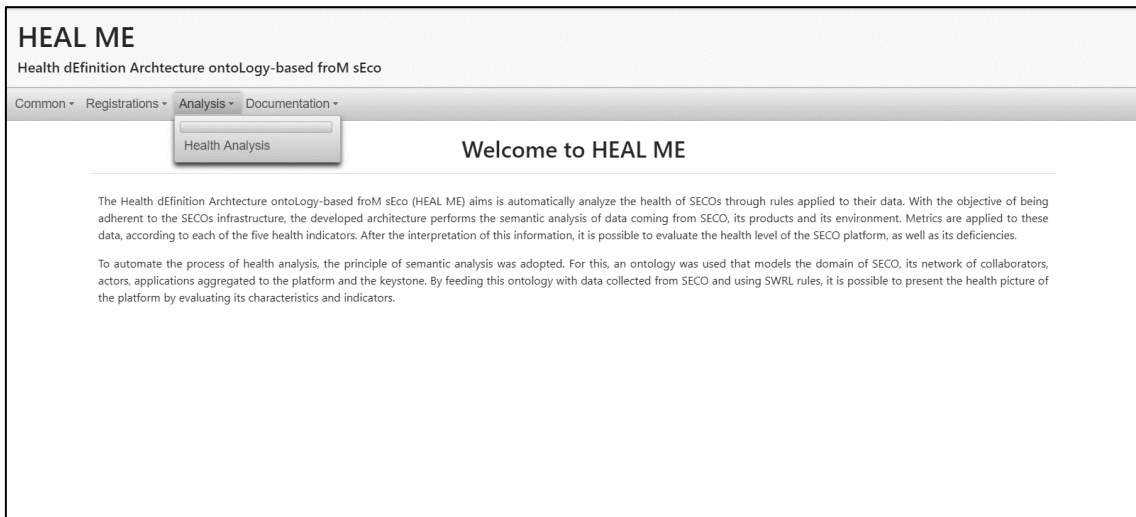


Figura 10: Acesso à página de análise.

Passo 3.1: Para disparar o processo de análise, basta selecionar a plataforma alvo na tela de seleção e após clicar no botão *Analysis* como demonstra a figura 11.

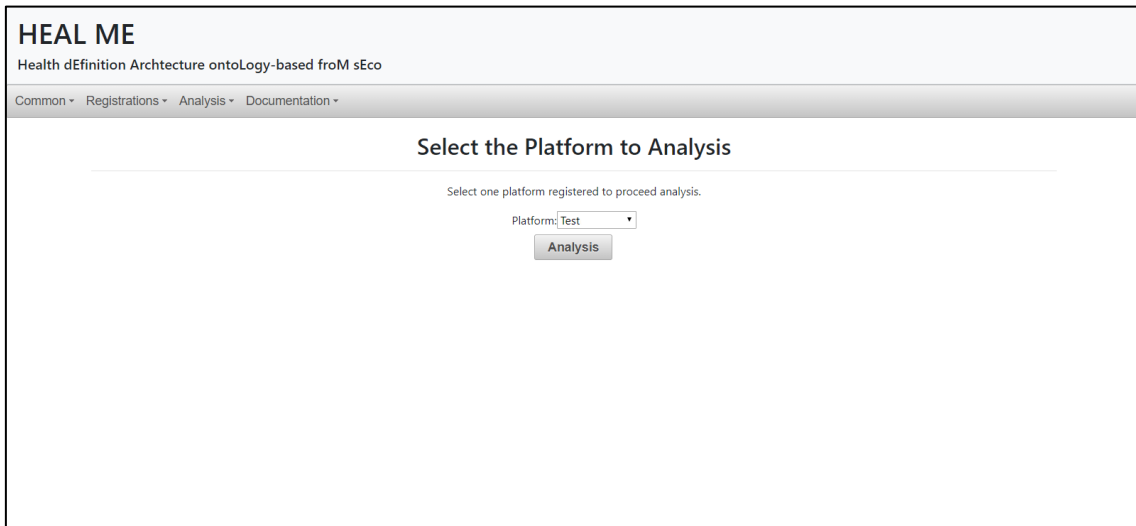


Figura 11: Seleção da plataforma para início do processo de análise.

Passo 4: Após executado o passo 3, os resultados do processamento da análise geram o relatório da análise, apresentado na página denominada *Analysis Overview*.

Passo 5: Através da interpretação dos gráficos exibidos é possível observar o panorama geral da saúde do ECOS analisado. Esta interpretação está descrita na próxima seção.

3. INTERPRETAÇÃO DOS RESULTADOS

A tela de resultados da análise é dividida em três setores. Estes são responsáveis por exibir o grau que o ECOS atende cada nível de granularidade dos componentes da análise. Esta granularidade é dividida em saúde geral, indicadores de saúde, características e métricas.

O primeiro setor é o *Analysis Overview*. O gráfico deste setor exibe a porcentagem atendida de cada indicador de saúde, a saber: sustentabilidade, diversidade, produtividade, robustez e criação de nicho. Em acréscimo, é exibido à esquerda o gráfico de saúde, demonstrando a porcentagem total de saúde atingida. A média das porcentagens dos indicadores compõe a porcentagem apresentada por este gráfico. Este setor é ilustrado pela Figura 12.

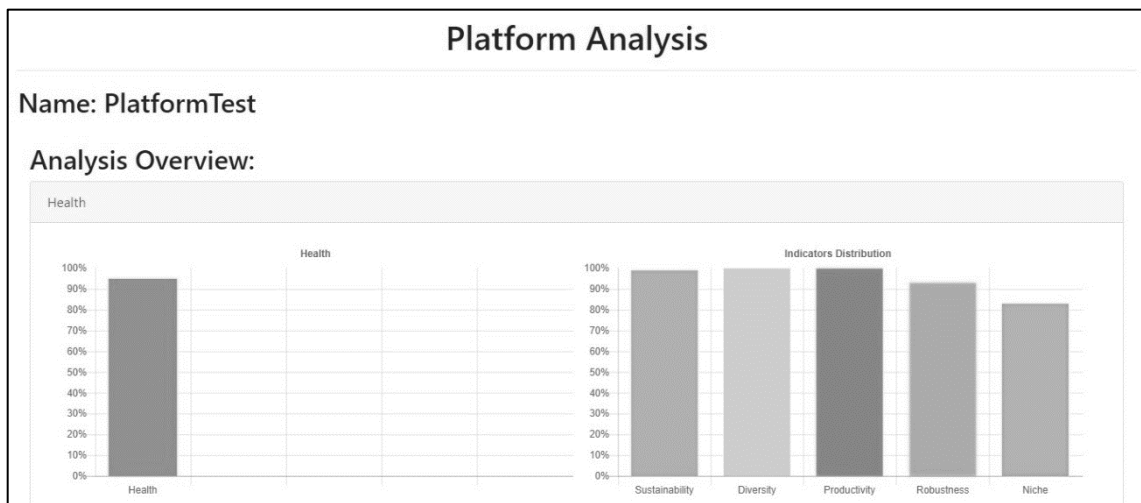


Figura 12: Health Overview.

O segundo setor é o *Individual Indicators*, ilustrado pela Figura 13. Este mostra a porcentagem das métricas atendidas que compõem cada uma das características de saúde individualmente. Estas características, por sua vez, compõem os indicadores. O indicador Sustentabilidade é composto pelas características Heterogeneidade, Habilidade de Regeneração, Balanceamento de Esforço, Balanceamento de Expertise e Visibilidade. Já o indicador Robustez é composto pelas características Interrelação, Consistência da Informação, Agrupamento e Consistência Financeira. Cada gráfico exibe a porcentagem de atendimento de seu indicador, calculado através da média das porcentagens de suas características. Os indicadores Diversidade, Produtividade e Criação de Nicho não são divididos em

características. Por este motivo, para cada um destes é exibido um gráfico com a porcentagem de atendimento das métricas que os compõem.

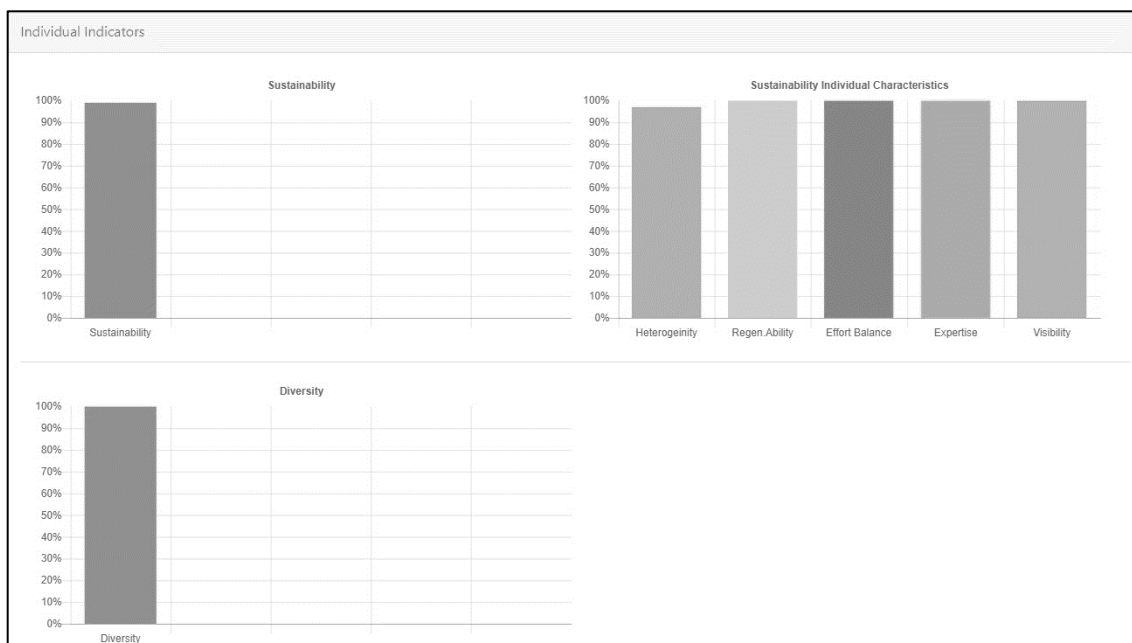


Figura 13: *Individual Indicators*.

Por fim, o ultimo setor mostra as métricas individualmente, agrupadas por característica ou indicador. As tabelas deste setor exibem o nome da métrica, o dado informado no cadastro da plataforma, o parâmetro correspondente e a porcentagem atingida pela medição.

O gráfico demonstra as métricas individualmente. Cada barra do gráfico representa uma métrica, e qual porcentagem a medição efetuada para aferi-la atingiu. No exemplo ilustrado na Figura 14 podemos visualizar a característica Heterogeneidade, que é composta pelas métricas Número de Países, Proximidade Semântica e Número de Tipos de Nós. No exemplo ilustrado, a métrica Número de Países atingiu 90% da medição em comparação ao parâmetro informado.

Caracteristic: Heterogeneity

Metric	Value	Parameter	(%)
Number of Countries	9	10	90.0%
Semantic Closeness	50	50	100.0%
Node Types	6	5	100.0%

Metrics Distribution:

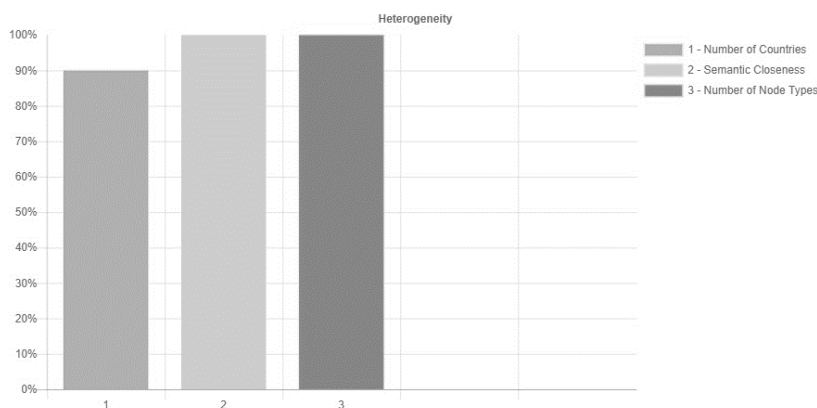


Figura 14: Gráfico individual da característica Heterogeneidade e suas respectivas métricas.

4. CONSIDERAÇÕES FINAIS

Utilizando a arquitetura Heal Me é possível visualizar o panorama geral de um ECOS e sua estrutura, baseando-se em dados sobre seu ambiente de desenvolvimento, usuários, apoio e visibilidade. Com a proposta desta arquitetura é analisar o ECOS com uma visão genérica, a percepção do usuário através da parametrização das análises é imprescindível. Desta forma, o usuário desta arquitetura poderá aproximar esta análise à realidade de seu ECOS, uma vez que cada um destes tem suas particularidades de tamanho, foco comercial e classificação da plataforma.