



SCENARI_{OT}CHECK: UMA TÉCNICA DE LEITURA BASEADA EM CHECKLIST PARA VERIFICAÇÃO DE CENÁRIOS IOT

Bruno Pedraça de Souza

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Guilherme Horta Travassos

Rio de Janeiro
Março de 2020

SCENARI_{IoT}CHECK: UMA TÉCNICA DE LEITURA BASEADA EM CHECKLIST PARA
VERIFICAÇÃO DE CENÁRIOS IOT

Bruno Pedraça de Souza

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO
LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS
NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM
ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Orientador: Guilherme Horta Travassos

Aprovada por: Prof. Guilherme Horta Travassos

Prof. Claudio Miceli de Farias

Prof^a. Flávia Coimbra Delicato

RIO DE JANEIRO, RJ - BRASIL

MARÇO DE 2020

Souza, Bruno Pedraça de

SCENARI_{IoT}CHECK: Uma Técnica de Leitura Baseada em Checklist para Verificação de Cenários IoT / Bruno Pedraça de Souza. – Rio de Janeiro: UFRJ/COPPE, 2020.

XII, 111 p.: il.; 29,7 cm.

Orientador: Guilherme Horta Travassos

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2020.

Referências Bibliográficas: p. 65 – 70.

1. Inspeção de Software. 2. Internet das Coisas. 3. Engenharia de Software Experimental. I. Travassos, Guilherme Horta. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Dedico este trabalho aos meus pais

Agradecimentos

Nesse curto caminho em que fiz o mestrado, conheci e tive inúmeros contatos com excelentes pesquisadores e amigos que se tornaram especiais a mim.

Primeiramente, gostaria de agradecer ao meu orientador Guilherme Horta Travassos pela sua disponibilidade, por todas as conversas, oportunidades, paciência, “puxões de orelha” e por me ajudar a me tornar um pesquisador. Sou eternamente grato por tudo que fez e continua fazendo por mim.

Aos meus pais, Dorian e Marcia, por todo amor, carinho e apoio dado a mim, e por terem ficado ao meu lado quando tomei a decisão de mudar para o Rio de Janeiro! Agradeço também ao Igor (meu “irmão” pet) e a toda minha família (tios, primos e padrinhos).

Aos professores Claudio Miceli e Flávia Delicato por aceitarem participar da minha banca e oferecerem valiosas contribuições. Muito obrigado!!!

Aos professores das disciplinas do PESC/COPPE por todo aprendizado passado a mim. Em especial aos professores: Ana Regina, Claudia Werner, Henrique Cukierman e Xexéo.

Aos amigos que fiz desde a época da graduação e que estão comigo até hoje, Andrews e Libni. Tivemos inúmeras conversas sobre mercado de trabalho e pelas trocas de experiências profissionais e acadêmicas. Obrigado também pelos conselhos durante essa etapa da minha vida!!! Gostaria de agradecer aos amigos Rafaella e Victor pela convivência enquanto compartilhávamos um apartamento e pela parceria de quase dois anos. Sei que não sou uma pessoa fácil de lidar. A Natalia, minha nova parceira e amiga, muito obrigado!

Aos amigos que fiz no Grupo de Engenharia de Software Experimental (ESE) – Alessandro, Andréa, Cecilia, Danyllo, Hélivio, Hilmer, Luciana, Rebeca, Talita, Taísa e a todo o grupo. Acredito que passei bons momentos com cada um. E vamos passar ainda mais!

Aos amigos já egressos do grupo ESE – Paulo Sergio, Victor Vidigal, Victor Machado e Valeria Silva. Tenho grande admiração por vocês. Obrigado por toda troca de experiência, pelas discussões sobre pesquisas científicas e sobre a vida.

Aos amigos que fiz durante os projetos que participei na COPPETEC e na empresa LEMOBS, obrigado a todos. Além disso, a todos os amigos que fiz na empresa LIQ. Obrigado pelo apoio!

Ao ICET/UFAM, em especial a Daniella de Oliveira pelas colaborações que tivemos juntos em minicursos e disciplinas. Aos participantes de todos os estudos experimentais.

A toda equipe da secretaria do PESC, Ricardo, Mercedes, Guty, Lurdes. A todos que sempre me atenderam de forma eficiente e gentil, meu muito obrigado!!!

Agradeço à CAPES e ao CNPq pelo apoio financeiro dado a mim e ao meu programa, sem o qual este trabalho não poderia ter sido realizado.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

SCENARI_{IoT}CHECK: UMA TÉCNICA DE LEITURA BASEADA EM CHECKLIST PARA VERIFICAÇÃO DE CENÁRIOS IOT

Bruno Pedraça de Souza

Março/2020

Orientador: Guilherme Horta Travassos

Programa: Engenharia de Sistemas e Computação

Sistemas de software no contexto da Internet das Coisas têm promovido uma revolução industrial, enfatizando novas características e preocupações como autonomia, conectividade contínua com dispositivos e interação entre sistemas, usuários e coisas. No entanto, a construção desse tipo de sistema de software ainda apresenta desafios, devido a suas particularidades. Estudos secundários evidenciam a falta de tecnologias para apoiar a construção desse tipo de sistema, na qual diferentes tecnologias e artefatos de software devem ser evoluídos ou criados para garantir a construção de soluções adequadas e com qualidade. A inspeção de software apoia a garantia da qualidade dos sistemas de software. Entretanto, até então não se tem conhecimento de técnica de inspeção aplicáveis a sistemas de software IoT. Nesse sentido, esta dissertação apresenta a SCENARI_{IoT}CHECK, uma técnica de inspeção baseada em *checklist* para apoiar a identificação de defeitos em cenários IoT. Os resultados dos estudos experimentais com a SCENARI_{IoT}CHECK indicam a viabilidade de sua utilização considerando a eficiência e efetividade observadas nos projetos nos quais foi aplicada.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

SCENARI_{IoT}CHECK: A CHECKLIST-BASED READING TECHNIQUE FOR
VERIFICATION OF IOT SCENARIOS

Bruno Pedraça de Souza

March/2020

Advisor: Guilherme Horta Travassos

Department: Computer Science and Systems Engineering

IoT software systems have directed the world into a current industrial revolution, bringing with itself new features and concerns such as autonomy, continuous device connectivity, and interaction among systems, users, and things. Nevertheless, to build these types of systems is still a problematic activity due to their specific features. Empirical studies show the lack of technologies to support the construction of this type of system, in which different software artifacts should be created to ensure their quality. Thus, software inspection has emerged as an alternative evidence-based method to support the quality assurance of artifacts produced during the software development cycle. However, there is no knowledge of inspection techniques applicable to IoT software systems. Therefore, this dissertation presents SCENARI_{IoT}CHECK, a checklist-based inspection technique to support the reading of IoT scenarios. SCENARI_{IoT}CHECK has been evaluated following an experimental methodology. The results of this research show that the technique has good results regarding cost-efficiency, efficiency, and effectiveness in the context of IoT system development.

Sumário

1	Introdução.....	1
1.1	Motivação e Contexto	1
1.2	Problema e Questão de Pesquisa.....	3
1.3	Objetivo.....	4
1.4	Metodologia	5
1.5	Contribuições e Publicações	6
1.6	Organização do Texto	7
2	Revisão Bibliográfica	9
2.1	Introdução.....	9
2.2	Cenários	10
2.3	Inspeção de Software	11
2.3.1	Processo de Inspeção de Software	12
2.3.2	Técnicas de Inspeção para Requisitos de Software	14
2.4	Internet das Coisas	16
2.5	Considerações Finais.....	19
3	Proposta Inicial da Técnica <i>SCENARI_{IoT}CHECK</i>	20
3.1	Introdução.....	20
3.2	<i>SCENARI_{IoT}</i> - uma técnica para especificação de cenários IoT	20
3.3	Facetas de IoT	23
3.4	Proposta Inicial da Técnica <i>SCENARI_{IoT}CHECK</i>	24
3.4.1	Mapeamento das Facetas IoT nos All	25
3.4.2	Técnica <i>SCENARI_{IoT}CHECK</i>	31
3.5	Considerações Finais sobre o Capítulo	33
4	Avaliando a Primeira Versão da <i>SCENARI_{IoT}CHECK</i>	34
4.1	Introdução.....	34

4.2 Estudo de Viabilidade	34
4.2.1 Planejamento	34
4.2.2 Execução	36
4.2.2.1 Estudo Piloto	36
4.2.2.2 Execução.....	36
4.2.2.3 Resultados da Caracterização.....	38
4.2.3 Resultados Qualitativos.....	38
4.2.4 Resultados Quantitativos.....	41
4.2.5 Ameaças a Validade e Limitação do Estudo.....	43
4.3 Evolução da Técnica SCENARI _{OT} CHECK.....	44
4.4 Considerações Finais sobre o Capítulo.....	45
5 Avaliando a Segunda Versão da SCENARI _{OT} CHECK.....	47
5.1 Introdução	47
5.2 Estudo de Observação.....	47
5.2.1 Planejamento	47
5.2.2 Execução	49
5.2.3 Resultados Quantitativos.....	51
5.2.4 Resultados Qualitativos.....	55
5.2.5 Ameaças a Validade	60
5.3 Considerações Finais sobre o Capítulo	61
6 Considerações Finais	62
6.1 Considerações Finais.....	62
6.2 Contribuições	63
6.3 Limitações da Pesquisa	63
6.4 Perspectivas Futuras	64
Referências	65
Apêndice A – Protocolo do Estudo de Viabilidade	71

Apêndice B - Primeira Versão da SCENARI _{OT} CHECK.....	76
Apêndice C - Registro de Inspeção – Versão 1.....	77
Apêndice D - Caracterização das questões da SCENARI _{OT} CHECK.....	78
Apêndice E – Questionário Pós-inspeção.....	91
Apêndice F – Protocolo do Estudo de Observação.....	93
Apêndice G - Segunda Versão da SCENARI _{OT} CHECK.....	100
Apêndice H - Registro de Inspeção – Versão 2.....	102
Apêndice I - Questionário Pós-inspeção – Versão 2.....	103
Apêndice J - Termo de Consentimento.....	106
Apêndice K - Caracterização do Participante.....	107
Apêndice L – Categorias das Questões do Checklist.....	109
Apêndice M – Template para Descrição de Cenários IoT.....	111

Lista de Figuras

FIGURA 1. METODOLOGIA UTILIZADA NESTA PESQUISA, ADAPTADA (SHULL, CARVER E TRAVASSOS 2001).....	5
FIGURA 2. ETAPAS DO PROCESSO DE INSPEÇÃO DE SOFTWARE (EXTRAÍDO DE (KALINOWSKI, TRAVASSOS, 2004))	12
FIGURA 3. ÁREAS DE APLICAÇÃO IOT	17
FIGURA 4. EXEMPLO DE UM AII SOBRE ATUAÇÃO ACIONADA POR UM INDIVÍDUO, COM BASE NOS DADOS DA IOT (EXTRAÍDO DE (SILVA, 2019)).....	21
FIGURA 5. EXEMPLO DE UM AII E SEU CATÁLOGO ASSOCIADO (EXTRAÍDO DE SILVA (2019))	22
FIGURA 6. FRAMEWORK PARA ENGENHARIA DE SISTEMAS BASEADOS EM IOT (EXTRAÍDO DE (MOTTA, DE OLIVEIRA, ET AL., 2019)).....	24
FIGURA 7. COLETA E EXIBIÇÃO DOS DADOS IOT PARA UM INDIVÍDUO	26
FIGURA 8. INDIVÍDUO ACIONANDO UMA ATUAÇÃO NO AMBIENTE.....	26
FIGURA 9. SISTEMA DE SOFTWARE ACIONANDO UMA ATUAÇÃO NO AMBIENTE	27
FIGURA 10. INDIVÍDUO ACIONANDO UMA ATUAÇÃO COM BASE EM DADOS DA IOT	28
FIGURA 11. SISTEMA DE SOFTWARE ACIONANDO UMA ATUAÇÃO COM BASE EM DADOS IOT. 28	
FIGURA 12. SISTEMA DE SOFTWARE ACIONANDO UMA ATUAÇÃO COM BASE EM DADOS NÃO IOT.....	29
FIGURA 13. SISTEMA DE SOFTWARE ACIONANDO UMA ATUAÇÃO NÃO IOT COM BASE EM DADOS IOT.....	30
FIGURA 14. INDIVÍDUO ACIONANDO UMA ATUAÇÃO COM BASE EM DADOS NÃO IOT.....	30
FIGURA 15. INDIVÍDUO ACIONANDO UMA ATUAÇÃO NÃO IOT COM BASE EM DADOS IOT	31
FIGURA 16. PROCESSO DE APLICAÇÃO DAS DUAS TÉCNICAS SCENARI _{IOT} E SCENARI _{IOT} CHECK.....	33
FIGURA 17. COMPREENSÃO DA SCENARI _{IOT} CHECK.....	39
FIGURA 18. UTILIDADE DA SCENARI _{IOT} CHECK	40
FIGURA 19. TRECHO DA EVOLUÇÃO DA TÉCNICA SCENARI _{IOT} CHECK.....	45
FIGURA 20. COMPORTAMENTO DAS TÉCNICAS AH E SC E AMBOS OS ESTUDOS DE VIABILIDADE (1. ESTUDO 1 – DESCRITO NO CAPÍTULO 4) E OBSERVAÇÃO (2. ESTUDO 2 – RODADA 1 DESCRITO NESSE CAPÍTULO) EM RELAÇÃO A EFICIÊNCIA	53
FIGURA 21. COMPORTAMENTO DAS TÉCNICAS AH E SC EM AMBOS OS ESTUDOS DE VIABILIDADE (ESTUDO 1) E OBSERVAÇÃO (ESTUDO 2) EM RELAÇÃO A EFICÁCIA	54

FIGURA 22. COMPORTAMENTO DAS TÉCNICAS AH E SC EM AMBOS OS ESTUDOS DE VIABILIDADE (ESTUDO 1) E OBSERVAÇÃO (ESTUDO 2) EM RELAÇÃO A CUSTO-EFICIÊNCIA	54
FIGURA 23. PERCEPÇÃO DOS PARTICIPANTES EM RELAÇÃO A DIFICULDADE DA APLICAÇÃO DA SCENARI _{IoT} CHECK.....	56
FIGURA 24. PERCEPÇÃO DOS PARTICIPANTES EM RELAÇÃO AS PERGUNTAS DA SCENARI _{IoT} CHECK.....	57
FIGURA 25. PERCEPÇÃO DOS PARTICIPANTES EM RELAÇÃO AO DESEMPENHO DELE EM APLICAR SCENARI _{IoT} CHECK.....	57

Lista de Tabelas

TABELA 1. STRING DE BUSCA DE TÉCNICAS DE INSPEÇÃO PARA REQUISITOS.....	12
TABELA 2. TAXONOMIA DE DEFEITOS.....	13
TABELA 3. PRINCIPAIS TÉCNICAS DE INSPEÇÃO PARA ARTEFATOS DE REQUISITOS	14
TABELA 4. STRING DE BUSCA PARA IOT.....	16
TABELA 5. EXEMPLO DE CARACTERIZAÇÃO DAS QUESTÕES PROVIDAS DOS ARRANJOS.....	32
TABELA 6. EXEMPLO DA CARACTERIZAÇÃO DAS QUESTÕES PROVIDAS DAS FACETAS IOT..	32
TABELA 7. CARACTERIZAÇÃO DE CADA GRUPO	38
TABELA 8. RESUMO DOS RESULTADOS OBTIDOS NO ESTUDO.	42
TABELA 9. RESUMO DOS INDICADORES DAS MEDIAS.....	42
TABELA 10. CARACTERIZAÇÃO DOS PARTICIPANTES.....	49
TABELA 11. DISTRIBUIÇÃO DOS PARTICIPANTES POR INSPEÇÃO.....	50
TABELA 12. MÉDIA DOS INDICADORES OBTIDOS	51
TABELA 13. PRIMEIRA RODADA DE INSPEÇÃO (UTILIZANDO TÉCNICA AH).....	52
TABELA 14. SEGUNDA RODADA DE INSPEÇÃO (UTILIZANDO A TÉCNICA SC).....	52
TABELA 15. TERCEIRA RODADA DE INSPEÇÃO (UTILIZANDO A TÉCNICA SC).....	53

1 Introdução

Neste capítulo são apresentadas a motivação e contexto para a realização deste trabalho, além dos objetivos e metodologia utilizada. E por fim, a organização dessa dissertação.

1.1 Motivação e Contexto

Atualmente, a sociedade vivencia e experimenta novas possibilidades de interação entre coisas, humanos, máquinas e sistemas. O objetivo é facilitar e melhorar a vida dos usuários finais (DESCHAMPS, LIAO, *et al.*, 2017). Com essas possibilidades de interação, os sistemas de software têm se tornado mais independentes para executar tarefas e, naturalmente, reduzindo a necessidade de intervenção humana. Isso leva a um cenário de trabalho inovador, configurando o que se identifica no campo como Quarta Revolução Industrial ou Indústria 4.0 (ATZORI, IERA, *et al.*, 2010, DESCHAMPS, LIAO, *et al.*, 2017). Sistemas autônomos, como os que envolvem o paradigma da Internet das Coisas (ou do inglês *Internet of Things* - IoT), sensibilidade ao contexto, características de ubiquidade e propriedades ciberfísicas estão contidos no cenário contemporâneo da indústria 4.0 (MOTTA, DE OLIVEIRA, *et al.*, 2018, SOUZA, MOTTA, *et al.*, 2019).

Sistemas de software IoT surgiram com uma série de características novas e desafios em sua construção (ZAMBONELLI, 2017, MOTTA, DE OLIVEIRA, *et al.*, 2018, SOUZA, MOTTA, *et al.*, 2019). Podemos tomar como exemplo os (MOTTA, DE OLIVEIRA, *et al.*, 2018). Motta *et al.* (2019) conduziram um estudo secundário com o propósito de identificar comportamentos e propriedades que os sistemas de software IoT possuem. Como resultado, observaram que sistemas de software IoT possuem 29 propriedades além de quatro comportamentos básicos: identificação, sensoriamento, processamento e atuação.

O comportamento de **identificação** é definido como a percepção dos objetos físicos (“coisas”) do mundo real por meio de algum tipo de identificador, como *tags*. No comportamento de **sensoriamento**, a captura dos dados é realizada por meio de algum tipo de sensor (*wearables* são considerados sensores nesse contexto). O **processamento** está associado a análise e interpretação dos dados obtidos pelo

sensoriamento. Por fim, **atuação** é a ação que o sistema realiza no ambiente em que está inserido.

Com melhor definição para a conexão das coisas, o termo IoT foi proposto e definido como um conjunto de tecnologias e dispositivos que são acessíveis, interagindo por meio de uma rede de comunicação (por exemplo, uma rede sem fio) e são capazes de capturar e trocar dados, comandos e tomar decisões, além de atuar em determinados contextos (MOTTA, R., DE OLIVEIRA, *et al.*, 2018, DE SOUZA, B, MOTTA, R, *et al.*, 2019). Diante deste cenário, a construção desses sistemas com qualidade ainda demanda um esforço inicial muito grande, devido às suas especificidades e comportamentos (ZAMBONELLI, 2017).

Atividades de garantia da qualidade são relevantes na engenharia de sistemas de software, pois contribuem para a melhoria e garantem que o processo de construção e os artefatos desenvolvidos atinjam os níveis de qualidade estabelecidos. Atividades de garantia de qualidade são normalmente associadas a técnicas de verificação e validação (V&V), que permitem identificar e, conseqüentemente, reduzir a propagação de defeitos ao longo do ciclo de desenvolvimento (BOEHM, 1984, AURUM, PETERSSON, *et al.*, 2002a) e contribuem de forma decisiva para aumentar a qualidade final do produto construído (AURUM, PETERSSON, *et al.*, 2002b, WALIA, CARVER, 2009).

Tendo em vista a necessidade de tornar disponível tecnologias de software para apoiar a construção de sistemas de software IoT com qualidade, este trabalho toma como ponto de partida os cenários IoT produzidos pela técnica *SCENARI_{IoT}* (SILVA, 2019) e apresenta uma solução tecnológica (técnica para inspeção de software) com base em técnicas de verificação e validação para garantir a qualidade desses cenários (WALIA, CARVER, 2009).

Inspeção de software é um método de verificação estático dos artefatos produzidos durante o ciclo de desenvolvimento de software que surgiu como uma subcategoria das revisões de software (FAGAN, 1986). As técnicas utilizadas nas inspeções de software normalmente são classificadas em três categorias: *ad-hoc*, baseadas em *checklist* e técnicas baseadas em leitura. (WALIA, CARVER, 2009, ALSHAZLY, ELFATATRY, *et al.*, 2014, EBAD, 2017).

A técnica de inspeção, *SCENARI_{IoT}CHECK*, proposta nesta dissertação é baseada em *checklist*, uma vez que este tipo de técnica possui perguntas gerais e específicas para verificar artefatos gerados durante o ciclo de desenvolvimento de

software, direcionando assim os inspetores à procura de defeitos específicos nesses artefatos. A *SCENARI_{IoT}CHECK* possui perguntas relacionadas as particularidades de sistemas de software IoT, o que torna o processo de inspeção mais eficiente e facilita a detecção de defeitos. Em geral, as perguntas do *checklist* podem ser originadas de revisões da literatura, projetos de software executados anteriormente, dentre outros (LAITENBERGER, 2002). A vantagem de utilizar a técnica baseada em *checklist* é poder guiar o inspetor na identificação de defeitos específicos no artefato (cenários) em revisão (FAGAN, 1986).

1.2 Problema e Questão de Pesquisa

A internet, antes utilizada apenas para interligar usuários a serviços remotos (*web* e *mobile*), passou a permitir a comunicação entre objetos e seres humanos com o propósito de oferecer serviços cada vez mais independentes. A evolução industrial permitiu a construção de novos sistemas, promovendo assim a evolução do paradigma da IoT (DESCHAMPS, LIAO, *et al.*, 2017).

A construção de sistemas de software tradicionais (i.e., *web*, *mobile*, dentre outros) ou contemporâneos (i.e., IoT, sensíveis ao contexto, dentre outros) requer um processo inicialmente bem definido, incluindo etapas como de especificação de requisitos até a implantação (MOTTA, DE OLIVEIRA, *et al.*, 2018).

Como se tem observado ao longo dos tempos, o sucesso dos projetos está diretamente associado a qualidade da especificação dos requisitos (GLASS, 1998). Diferentes tecnologias para apoiar a especificação de requisitos com qualidade estão disponíveis para sistemas de software convencionais. Entretanto, para sistemas de software contemporâneos, ainda não se tem um conjunto de tecnologias que atendam as expectativas de construção, ou que tenham tido sua utilização avaliada em projetos de sistemas de software contemporâneos, tornando, por exemplo, desafiador escolher quais tecnologias, métodos e técnicas são mais apropriadas para sua construção e garantia da qualidade (ZAMBONELLI, 2017, MOTTA, SILVA, *et al.*, 2019).

Neste sentido, algumas abordagens têm sido adaptadas, avaliadas e propostas para apoiar a construção dos sistemas baseados em IoT. Podemos citar, por exemplo, técnicas para elicitare e especificar requisitos baseados em cenários específicos de sistemas IoT (SILVA, 2019), técnicas adaptadas para especificar requisitos IoT através de Use Cases (AZIZ, SHEIKH, *et al.*, 2016, REGGIO, 2018), frameworks para

descrever requisitos não-funcionais de segurança de sistemas IoT (SAHA, AUDDY, *et al.*, 2017), dentre outros.

Embora se observe a existência de algumas propostas de abordagens para apoiar a elicitaco e especificaco de requisitos de sistemas de software IoT, pouco se encontra na literatura tcnica sobre tecnologias que permitam garantir a qualidade dos requisitos especificados, tais como tcnicas de reviso ou inspeco (SOUZA, MOTTA, *et al.*, 2019). Por exemplo, a *SCENARI_{IoT}*  uma tcnica de elicitaco e especificaco para descrio de cenrios baseados em IoT (SILVA, 2019). A tcnica adapta e evolui tecnologias convencionais de cenrios de software para apoiar a especificaco de cenrios IoT. Apesar das vantagens obtidas com seu uso, a *SCENARI_{IoT}* no oferece facilidades para a garantia da qualidade (consistncia e cobertura) dos requisitos descritos. Visando a contribuir com a evoluo da *SCENARI_{IoT}*, esta dissertaco responde  questo de pesquisa: *Como realizar a inspeco de requisitos de sistemas de software IoT para verificar se os componentes e propriedades especificados pela SCENARI_{IoT} foram capturados/descritos adequadamente nos cenrios?*

1.3 Objetivo

O objetivo deste trabalho  desenvolver uma tecnologia de verificaco de software que permita a inspeco de artefatos de cenrios baseados em sistemas IoT. Especificamente, o pressuposto desta pesquisa  definir uma tcnica de inspeco baseada em *checklist* para apoiar a identificaco de defeitos no artefato produzido na especificaco de requisitos de sistemas de software IoT.

Dessa forma, este trabalho vem como um esforo inicial de propor, adotar e inserir tecnologias de verificaco de artefatos de requisitos de software, especificamente para cenrios desenvolvidos com a *SCENARI_{IoT}*, no contexto de sistemas de software IoT. Seus objetivos especficos so descritos a seguir:

- Definir as caractersticas de cenrios e sistemas IoT (por meio de estudos secundrios) e capturadas pela *SCENARI_{IoT}* que devem ser verificadas.
- Desenvolver uma tecnologia de software (tcnica de inspeco baseada em *checklist*, *SCENARI_{IoT}CHECK*) para avaliao e verificaco de cenrios baseados em IoT produzidos pela *SCENARI_{IoT}*.
- Avaliar a tcnica de inspeco *SCENARI_{IoT}CHECK* desenvolvida por meio de estudos experimentais.

1.4 Metodologia

A proposta do método de desenvolvimento utilizada nesta pesquisa é uma adaptação da metodologia proposta por Shull, Carver e Travassos (2001). Esta metodologia é baseada em evidência e consiste na realização de uma série de estudos experimentais para a maturação da tecnologia a ser concebida, construída e, posteriormente, adotada pelos engenheiros de software. A Figura 1 mostra os estágios previstos no método de construção da *SCENARI_{OT}CHECK*.

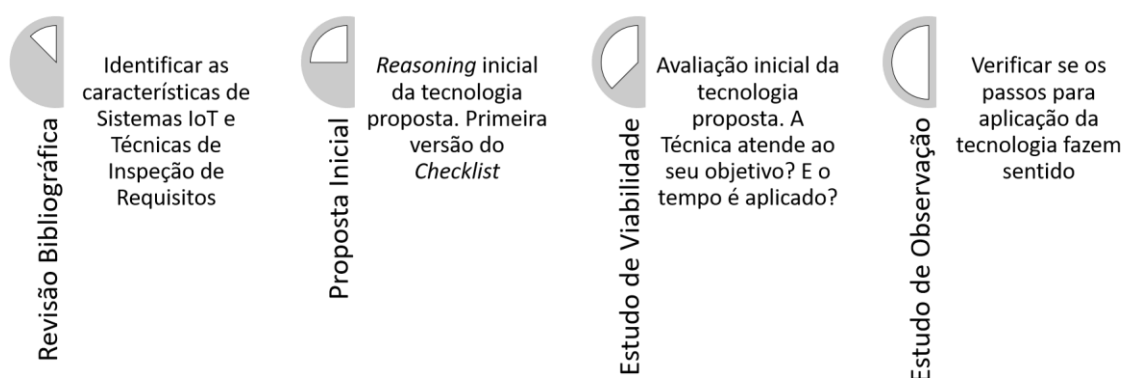


Figura 1. Metodologia utilizada nesta pesquisa, adaptada (SHULL, CARVER e TRAVASSOS 2001)

- *Revisão Bibliográfica*: em busca de estudos secundários com o objetivo de obter as principais técnicas de inspeção de requisitos utilizadas no contexto de sistemas convencionais e contemporâneos, bem como suas características e definições. Além disso, buscar por estudos secundários que tratem de características, propriedades e definições de sistemas de software IoT.
- *Proposta Inicial*: definição de um conjunto de questões para o *checklist* inicial da *SCENARI_{OT}CHECK* de acordo com a revisão bibliográfica realizada e seus resultados encontrados.
- *Estudo de Viabilidade*: com o propósito de avaliar se a técnica *SCENARI_{OT}CHECK* alcança seu objetivo de detectar defeitos nos cenários produzidos pela técnica de especificação *SCENARI_{OT}*. Para realizar esta etapa, um estudo experimental foi realizado com ambas as técnicas *SCENARI_{OT}* e *SCENARI_{OT}CHECK*.
- *Estudo de Observação*: com o propósito de aprimorar a abordagem proposta em relação a compreensão da sua aplicação. Além de avaliar a eficiência e eficácia

da técnica *SCENARI_{IoT}CHECK*. Neste estudo, as técnicas *SCENARI_{IoT}* e *SCENARI_{IoT}CHECK* foram utilizadas de forma integrada.

1.5 Contribuições e Publicações

Diferentes contribuições podem ser observadas com a leitura dessa dissertação. De forma objetiva, as principais contribuições podem ser apresentadas como:

- Apresentar um sumário das técnicas de inspeção para requisitos descritas na literatura técnica e evidenciadas através de estudos secundários.
- Organizar uma nova técnica de inspeção baseada em *checklist* (*SCENARI_{IoT}CHECK*) para revisar cenários específicos para o domínio de sistemas de software IoT.
- Avaliar, experimentalmente, o *checklist* e apresentar indícios sobre sua capacidade em apoiar a garantia da qualidade de artefatos de sistemas de software IoT.

Ao longo da realização deste trabalho, algumas publicações diretamente relacionadas à construção da *SCENARI_{IoT}CHECK* foram preparadas:

- **SOUZA, B. P.; MOTTA, R. C.; TRAVASSOS, G. H. The first version of SCENARI_{IoT}CHECK: A Checklist for IoT based Scenarios.** In: XXXIII Brazilian Symposium on Software Engineering (SBES), Salvador, 2019, p. 219-223.
- **SOUZA, B. P.; MOTTA, R. C.; COSTA, D. O.; TRAVASSOS, G. H. An IoT-based Scenario Description Inspection Technique.** In: XVIII Brazilian Symposium on Software Quality (SBQS), 2019, Fortaleza. XVIII Brazilian Symposium on Software Quality (SBQS), 2019. v. 18.
- **SOUZA, B. P.; MOTTA, R. C.; TRAVASSOS, G. H. Towards the Description and Representation of Smartness in IoT Scenarios Specification.** In: XXXIII Brazilian Symposium on Software Engineering (SBES), 2019, Salvador. (*Melhor artigo* - XXXIII SBES – Trilha de *Insightful Ideas & Emerging Results*).

- SILVA, D. V.; **SOUZA, B. P.**; GONÇALVES, T. G.; TRAVASSOS, G. H. **Uma Tecnologia para Apoiar a Engenharia de Requisitos de Sistemas de Software IoT.** In: XXIII Congresso Ibero-Americano de Engenharia de Software (CibSE), Curitiba, 2020.

Adicionalmente, a participação em atividades de projetos de pesquisa em engenharia de software permitiu que experiências adicionais fossem discutidas, das quais se destacam:

- DOS SANTOS, P. S. M.; BELTRÃO, A. C.; **DE SOUZA, B. P.**; TRAVASSOS, G. H. **On the benefits and challenges of using kanban in software engineering: a structured synthesis study.** JOURNAL OF SOFTWARE ENGINEERING RESEARCH AND DEVELOPMENT, v. 6, p. 01-29, 2018.
- **SOUZA, B. P.**; DORESTE, A.; REIS, C.; XEXEO, G. **Utilizando Framework MDA para Avaliar a Estética de um Jogo: Um Estudo Preliminar sobre a Percepção de Estudantes de Graduação.** In: XVII Simpósio Brasileiro de Games e Entretenimento Digital (SBGames), 2018, Foz do Iguaçu. V. 1. p. 348-351.
- COSTA, D. S.; COSTA, D. O.; BONIFACIO, B. A.; **SOUZA, B. P.**; FERNANDES, P. S. **Using Frameworks for Rapid Applications Development as Learning Object for Teaching Web Programming.** In: XIII Latin American Conference on Learning Technologies (LACLO), 2018, São Paulo.
- COSTA, A. L. B.; BONIFACIO, B. A.; **SOUZA, B. P.**; FERNANDES, P. S. **Applying Teaching Methods as Requirements to Develop Games to Assist Children with Learning Disabilities: A Case Study.** In: Latin American Symposium on Computing and Society (CLEI), 2018, São Paulo.

1.6 Organização do Texto

Esta dissertação está organizada em outros cinco capítulos, além deste primeiro que descreveu a introdução, motivação e o contexto no qual essa dissertação está inserida. A organização desse trabalho segue a estrutura abaixo:

Capítulo 2 – Revisão Bibliográfica: Descreve a revisão realizada em busca dos conceitos relacionados à Inspeção de Software e Internet das Coisas (IoT).

Capítulo 3 – Proposta Inicial da Técnica SCENARI_{IoT}CHECK: Apresenta a proposta da técnica *SCENARI_{IoT}CHECK – A Checklist for Reading IoT-based Scenarios*, uma técnica de inspeção para avaliação de cenários baseados em sistemas de software IoT. Esta técnica combina perspectivas das abordagens *SCENARI_{IoT}* e Facetas IoT.

Capítulo 4 – Estudo de Viabilidade: Descreve o estudo executado com o propósito de verificar se a técnica *SCENARI_{IoT}CHECK* alcança seu objetivo de identificar defeitos, além de buscar subsídios para sua evolução.

Capítulo 5 – Estudo de Observação: Descreve o estudo realizado com a segunda versão da técnica *SCENARI_{IoT}CHECK* a fim de compreender como os inspetores aplicam a técnica.

Capítulo 6 – Considerações Finais e Perspectivas Futuras: Descreve as conclusões e contribuições desse trabalho, além de mostrar as perspectivas futuras da pesquisa em questão.

2 Revisão Bibliográfica

Neste capítulo, são abordados os principais conceitos sobre inspeção de software, como as técnicas existentes e disponíveis na literatura, além de tratar sobre suas aplicações. Também são destacadas algumas definições de Internet das Coisas (IoT) e suas aplicações.

2.1 Introdução

A engenharia de software é uma disciplina que envolve colaboração (teoria) e aplicação (prática), contribuindo diretamente para a qualidade do produto construído. Qualidade de software é um requisito fundamental para que o produto produzido tenha uma boa aceitação pelos usuários e está relacionada à capacidade de um produto ou serviço em atender as necessidades dos usuários finais (CONTE, TRAVASSOS, 2009).

Para garantir a qualidade do produto, Boehm (1984) definiu dois modos de explorar se um produto está sendo construído de maneira correta, são elas: verificação (se os requisitos estão em conformidade com a construção do sistema) e validação (garantir que o produto gerado corresponda as necessidades do usuário).

“Don't worry about that specification paperwork. We'd better hurry up and start coding, because we're going to have a whole lot of debugging to do” (BOEHM, 1984).

Nesse contexto, realizar as tarefas de verificação e validação (V&V) são essenciais para obter a qualidade final de um produto de software. Essas técnicas são aplicadas durante todo o processo de desenvolvimento do software com o intuito de revisar todos os artefatos produzidos e verificar suas respectivas conformidades, consistência e qualidade (FAGAN, 1986). As propostas de técnicas de revisão de software, para serem aplicadas nas fases iniciais do processo de construção do produto, permitiram reduzir o retrabalho e a propagação de inconformidades (defeitos) para outras fases do ciclo de desenvolvimento (WALIA, CARVER, 2009, ALSHAZLY, ELFATATRY, *et al.*, 2014, EBAD, 2017).

A inspeção de software é uma técnica de verificação de artefatos de software, que surgiu como um método formal que visa examinar os artefatos produzidos nas

diferentes etapas do ciclo de desenvolvimento do software. Seu principal objetivo é detectar defeitos nas fases iniciais do projeto (FAGAN, 1986, AURUM, PETERSSON, *et al.*, 2002, WALIA, CARVER, 2009).

Para dar continuidade na discussão dos conceitos envolvidos no desenvolvimento desse trabalho, a Subseção 2.2 apresenta o conceito de cenários utilizado nessa pesquisa. Uma revisão da literatura sobre Inspeção de software é apresentada na Seção 2.3, a qual discute definições, técnicas encontradas e processo de inspeção de software. Em seguida, a Seção 2.4 descreve os principais conceitos sobre IoT e suas características.

2.2 Cenários

De acordo com a IEEE (1998), um requisito é definido como “a capacidade ou condição necessária para que um usuário resolva um problema ou alcance um objetivo”. Além disso, um requisito especificado é uma descrição da funcionalidade de um sistema de software. Os requisitos de software podem ser interpretados e extraídos do mundo real, da necessidade do usuário, ou pode ser derivado da análise técnica de um sistema computacional.

Atualmente existem uma variedade de pesquisas, métodos, técnicas e *templates* para realizar a especificação de requisitos de maneira eficiente, eficaz e adequada (DAR, LALI, *et al.*, 2018, PACHECO, GARCIA, *et al.*, 2018, LIM, CHUA, *et al.*, 2019). No entanto, especificar requisitos é uma tarefa difícil, em que muitos engenheiros de software têm dificuldade ao abstrair o que o cliente deseja, assim, especificando requisitos inadequados para o problema proposto.

A elaboração da especificação de requisitos é o primeiro passo do ciclo de construção de um sistema. É um passo essencial, pois especificar requisitos de maneira inadequada pode levar a problemas de desenvolvimento do produto no futuro (GLASS, 1998). Para mitigar a especificação de requisitos inadequados, métodos e técnicas são utilizados para auxiliar na especificação de requisitos, como entrevistas, questionários, *brainstorms*, cenários e outros.

Na fase de especificação de requisitos, os cenários são amplamente utilizados pelos profissionais e pesquisadores de engenharia de software, pois é uma técnica muito difundida, simples de aplicar, além de ser uma técnica já consolidada na literatura (MAIDEN, 1998, LEITE, DOORN, *et al.*, 2005, NARDI, 2007).

Cenário é uma técnica de especificação de requisitos e tem o propósito de descrever um possível conjunto de eventos que podem ocorrer no ambiente (CARROLL, 1995, LEITE, DOORN, *et al.*, 2005). O principal objetivo do desenvolvimento de cenários é estimular o pensamento sobre possíveis ocorrências, suposições relacionadas sobre essas ocorrências, possíveis oportunidades, riscos e cursos de ação que podem ocorrer no sistema (LIU, YU, 2004). Além disso, consiste em uma coleção de narrativas (ações e eventos) de situações que podem acontecer no ambiente, ajudando a coletar informações, identificar problemas e antecipar soluções (LEITE, DOORN, *et al.*, 2005). Uma vantagem do uso de cenários é que estes são um meio de representação de fácil compreensão para os clientes, os usuários envolvidos e para o time de desenvolvimento. A descrição dos cenários pode ser feita de várias formas, entre elas: *mockups*, *storyboards* e casos de uso (SILVA, 2019).

2.3 Inspeção de Software

A inspeção de software é uma técnica estática de verificação de software que surgiu como uma subcategoria de revisões de software. É muito aplicada por pesquisadores e profissionais da prática da área de engenharia de software (FAGAN, 1986, LEITE, DOORN, *et al.*, 2005, DOS SANTOS, TRAVASSOS, 2010).

Com o objetivo de detectar defeitos e, conseqüentemente, melhorar os artefatos produzidos pelo ciclo de desenvolvimento de um software, as técnicas de inspeção visam realizar revisões em artefatos desde as primeiras fases (especificação de requisitos) do ciclo de desenvolvimento até a última (teste de software) (BOEHM, 1984).

De acordo com Basili, Green, *et al.* (1996), é recomendável que a inspeção seja aplicada nas fases iniciais do projeto, visto que, quanto mais cedo os defeitos forem detectados, menos custoso e mais fácil será para corrigi-los. Pesquisas com aplicação de inspeções têm se mostrado eficientes, eficazes e com baixo custo para encontrar defeitos em artefatos.

Atualmente, existem diferentes técnicas de inspeção de software específicas para cada etapa do processo de desenvolvimento e seus respectivos artefatos. Por exemplo, técnicas para avaliação de requisitos (FAGAN, 1986, BASILI, GREEN, *et al.*, 1996), usabilidade (CONTE, MASSOLLAR, *et al.*, 2007), diagramas de UML

(TRAVASSOS, SHULL, *et al.*, 1999, NAKAZATO e TRAVASSOS, 2013), verificação de documentos de casos de testes (DE BRITO COSTA, 2012), dentre outras.

Visando identificar técnicas de inspeção disponíveis na literatura técnica, realizamos a busca por estudos secundários que definissem e tratassem de técnicas de inspeções. Para isso, foram utilizados os termos de busca disponíveis na Tabela 1. A *Scopus*¹ foi utilizada como máquina de busca, uma vez que ela indexa várias outras máquinas de busca. A pesquisa selecionou os trabalhos sobre inspeção entre os anos de 2009 a 2019.

Tabela 1. String de busca de técnicas de inspeção para requisitos

Busca	String
Técnica de Inspeção de Requisitos	(("systematic literature review" OR "systematic mapping" OR "literature review") AND ("software inspection") AND requirements)

De acordo com os resultados, quatro estudos secundários foram encontrados (WALIA, CARVER, 2009, ALSHAZLY, ELFATATRY, *et al.*, 2014, HERNANDES, BELGAMO, *et al.*, 2014, EBAD, 2017). Todos os estudos encontrados definem e caracterizam técnicas de inspeção de software específicas para requisitos. A subseção seguinte descreverá o processo de aplicação de uma inspeção de software.

2.3.1 Processo de Inspeção de Software

Para realizar a inspeção de software, é necessário seguir uma série de passos definidos, como: escolher qual artefato será inspecionado, as pessoas que irão realizar a inspeção, além do planejamento e roteiro da inspeção. De acordo com (FAGAN, 1986), o processo de inspeção segue seis etapas, como mostra a Figura 2.

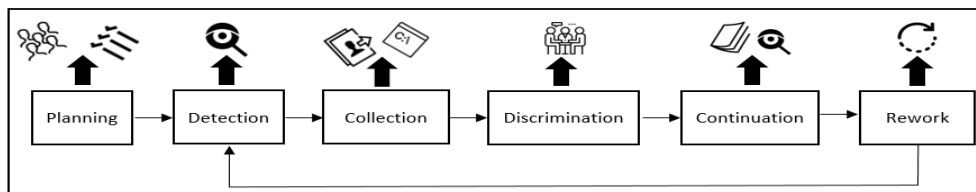


Figura 2. Etapas do processo de inspeção de software (extraído de (KALINOWSKI, TRAVASSOS, 2004))

¹ www.scopus.com

- i. *Planejamento*: seleção da equipe de inspeção, assim como a definição dos papéis para cada membro da equipe. Nesta etapa, também são selecionados os artefatos a serem inspecionados, datas são definidas e reuniões são marcadas para outras etapas.
- ii. *Detecção*: nesta etapa, os inspetores revisam o artefato, identificando e relatando discrepâncias² (possíveis defeitos).
- iii. *Coleção*: agrupamento das discrepâncias relatadas pelos inspetores após a finalização da inspeção.
- iv. *Discriminação*: nesta etapa, são discutidas as discrepâncias detectadas após a inspeção. O moderador avalia junto ao grupo o que é defeito e o que é falso positivo³ e duplicatas⁴. Após a classificação da discrepância, é realizada a junção dos defeitos reais numa lista para a eventual correção. Cada defeito encontrado e listado é classificado de acordo com a taxinomia utilizada pela equipe. Um relatório também é escrito nesta etapa.
- v. *Retrabalho*: após a etapa de discriminação e gerada a lista com os defeitos encontrados, o autor (desenvolvedor, analista, engenheiro de software, testador) corrige o artefato produzido na discriminação.
- vi. *Continuação*: o moderador verifica cada parte do novo artefato produzido e decide se é necessária uma nova inspeção ou não.

Além de definir o processo de inspeção utilizado (SAUER, ROSS JEFFERY, *et al.*, 2000, KALINOWSKI, TRAVASSOS, 2004), a taxonomia de defeitos para classificação dos defeitos encontrados também foi definida (SHULL, 1999). A Tabela 2 descreve os tipos de defeitos com suas respectivas definições. Esta taxonomia é utilizada nos estudos desta dissertação.

Tabela 2. Taxonomia de defeitos (extraída de (SHULL, 1999))

Defeito	Descrição Geral
Omissão	Informação importante que deveria ter sido especificada no artefato, porém foi omitida no mesmo.
Fato incorreto	Informações no artefato de software contradizem informações no documento requisitos ou no conhecimento de domínio geral.
Inconsistência	Informações dentro de uma parte dos artefatos de software são inconsistentes com outras informações no artefato de software.
Ambiguidade	A informação contida no artefato é ambígua, ou seja, passível de mais de

² Discrepâncias - são os possíveis defeitos que foram identificados pelos inspetores. Porém, a discrepância só é classificada como defeito após a reunião de discriminação.

³ Falso positivo - são discrepâncias que foram detectadas no artefato, porém, não são classificadas como defeito, após a reunião de discriminação.

⁴ Duplicatas: mesmo defeito encontrado repetidas vezes no mesmo artefato e reportado por diferentes inspetores.

	uma interpretação.
Informação Estranha	Informação é fornecida, no entanto ela não é necessária ou utilizada no sistema.

2.3.2 Técnicas de Inspeção para Requisitos de Software

Walia e Carver (2009) realizaram uma revisão sistemática com objetivo de identificar os tipos de erros de requisitos de forma que pudesse melhorar a qualidade do artefato do software produzido no ciclo de desenvolvimento. Como resultado do trabalho, técnicas de inspeção de requisitos foram encontradas na revisão sistemática e estão descritas de acordo com a Tabela 3. Vale ressaltar que extraímos apenas as técnicas específicas que inspecionassem artefatos de requisitos. Excluimos todas as técnicas que estavam fora do escopo desta pesquisa, como técnicas para verificar código, casos de testes e modelos UML.

Tabela 3. Principais técnicas de inspeção para artefatos de requisitos

Técnica de Inspeção	Foco de aplicação	Referência
<i>Ad-Hoc technique</i>	Todos os documentos	-
<i>Checklist Based Reading technique</i>	Especificações de requisitos	Fagan (1976)
<i>Defect Based Reading technique</i>	Especificações de requisitos	Porter and Votta (1994)
<i>Usage Based Reading technique</i>	Especificações de requisitos	Thelin et al. (2004)
<i>Function point Reading technique</i>	Especificações de requisitos	Cheng and Jeffery (1996)
<i>Traceability Based Reading Technique</i>	Requisitos, documentos de design	Travassos et al. (1999)
<i>Perspective Based Reading technique</i>	Requisito, documentos de design, módulos de código funcional, artefatos OO	Basili et al. (1996)
<i>Metric-Based Reading technique</i>	Inspeção de Requisitos	Bernárdez et al. (2004)
<i>Inspection using Error Abstraction</i>	Documentos de requisitos	Lanubile et al. (1998)
<i>N-Fold Inspection Process</i>	Documento de requisitos do usuário	Martin and Tsai (1990)

O resumo das técnicas encontradas por Walia e Carver (2009) e pela revisão bibliográfica *ad-hoc* são descritos a seguir:

- **Ad-Hoc technique** – é a base para a criação de outras técnicas de inspeções. Não existe uma padronização na sua aplicação. É puro conhecimento tácito do inspetor na detecção de defeitos.

- **Checklist Based Reading technique** – é um conjunto de perguntas fornecidas para apoiar o inspetor na detecção de defeitos durante a leitura do documento do software. É uma técnica que necessita de conhecimento prévio sobre o artefato que está sendo inspecionado (FAGAN, 1986).
- **Defect Based Reading technique** – fornece aos revisores diferentes classes de falhas ao invés de fornecer uma lista de perguntas. Os revisores inspecionam os documentos em busca de defeitos nessas classes. É aplicado com sucesso às especificações de requisitos (PORTER, VOTTA, 1994).
- **Usage Based Reading technique** – utiliza casos de uso para ajudar o revisor a inspecionar o documento de requisitos do ponto de vista do usuário. Ele ajuda o revisor a se concentrar mais profundamente e detectar falhas que são mais críticas (THELIN, RUNESON, *et al.*, 2002).
- **Function point Reading technique** – é outra variante de técnicas de leitura baseadas em cenários. Envolve a construção de questões relacionadas ao domínio para todas as questões-chave, complementadas por lista de verificação e é usada para desenvolver cenários para investigar diferentes aspectos da SRS (CHENG, JEFFERY, 1996).
- **Traceability Based Reading Technique** – é uma técnica usada para verificar a consistência entre o projeto e os documentos de requisitos (leitura vertical) e para garantir a correção interna do projeto (leitura horizontal) (TRAVASSOS, SHULL, *et al.*, 1999).
- **Perspective Based Reading technique** – é uma técnica de leitura baseada em cenários que fornece orientação processual, adaptada aos requisitos da linguagem natural. Todo o processo envolve a seleção de um conjunto de perspectivas, adequando os procedimentos para cada perspectiva com o objetivo de construir uma abstração dos requisitos. Em seguida, descreve as questões para encontrar defeitos em cada modelo e os revisores o usam para revisar o documento (BASILI, GREEN, *et al.*, 1996).
- **Metric-Based Reading technique** - envolve a detecção de tipos específicos de defeitos em artefatos de software de linguagem natural usando um conjunto de regras ou heurísticas que descrevem a relação entre os valores da métrica e a presença de certos tipos de defeitos. Os inspetores usam essas heurísticas ou um conjunto de regras para detectar defeitos. Experimentos mostram que

essa abordagem ajuda a detectar mais defeitos do que a revisão de listas de verificação (BERNÁRDEZ, GENERO, *et al.*, 2004).

- **Inspection using Error Abstraction** – essa técnica envolve a detecção de falhas usando uma taxonomia de falhas, seguida pela abstração das causas das falhas detectadas na primeira etapa e, em seguida, pelo uso das informações de erro para inspecionar novamente o documento e localizar mais falhas examinadas pela primeira vez. Apoiar-se fortemente na criatividade e capacidade dos revisores de abstrair causas de falhas (LANUBILE, SHULL, *et al.*, 1998, LANUBILE, SHULL, *et al.*, 2002).
- **N-Fold Inspection Process** – esta técnica emprega o processo de inspeção formal replicado em paralelo usando N equipes independentes. Baseia-se na hipótese de que há pouca sobreposição entre as falhas encontradas por cada equipe paralela e cada equipe detecta diferentes falhas (MARTIN, TSAI, 1990).

2.4 Internet das Coisas

Internet das coisas é comumente conhecida pela sigla IoT na academia e na indústria. Sistemas IoT são conceituados como objetos interconectados entre si por meio da internet, que permitem qualquer tipo de conexão em qualquer lugar e a todo instante (ATZORI, IERA, *et al.*, 2010). De acordo com Giusto *et al.* (2010), a abstração básica deste paradigma é a presença de uma grande variedade de objetos - como RFID, NFC, sensores, atuadores, celulares, dentre outros, que, por meio de esquemas de endereçamento únicos (*tags*), são capazes de interagir uns com os outros.

De acordo com Chen, Xu, *et al.* (2014), no contexto do paradigma de sistemas IoT, diversos objetos ao nosso redor são conectados a uma rede sem fio de uma forma ou de outra. A identificação de RFID e *tags* a tecnologias de sensores e outras tecnologias inteligentes são incorporadas a uma variedade de aplicações.

Nesta dissertação, os termos utilizados para realizar a busca de estudos secundários sobre IoT estão disponíveis na Tabela 4. A *Scopus* também foi utilizada como máquina de busca.

Tabela 4. String de busca para IoT

Busca	String
Internet das Coisas	((("systematic literature review" OR "systematic mapping" OR "literature review") AND ("internet of things" OR "IoT"))

2.4.1. Aplicação de Sistemas IoT

O uso de sistemas de software IoT têm aumentado muito nos últimos anos, devido a sua popularidade em tornar os sistemas mais inteligentes, autônomos e tendo um baixo custo de construção (ATZORI, IERA, *et al.*, 2010, MOTTA, SILVA, *et al.*, 2019). Esses sistemas estão contidos em diversas áreas da vida dos usuários, como agricultura, saúde, lazer, cidades inteligentes, dentre outros (ATZORI, IERA, *et al.*, 2010, CHEN, XU, *et al.*, 2014, MOTTA, SILVA, *et al.*, 2019).



Figura 3. Áreas de aplicação IoT

As cidades inteligentes surgiram com o propósito de promover o bem-estar da população e ao mesmo tempo construir o crescimento econômico associado à sustentabilidade, acessibilidade e gerando riquezas e empregos para a população. As aplicações tecnológicas para as cidades são pensadas e criadas utilizando o paradigma IoT para automatizá-las. A definição de Cidade Inteligente apresentada por Gaur *et al.*, (2015), é um ecossistema urbano complexo e emaranhado, onde são incorporando diversos outros sistemas complexos de infraestrutura, comportamento humano, tecnologia, estruturas sociais, políticas e econômica. Dentro de uma cidade inteligente, o gerenciamento de componentes é feito levando em consideração seus principais pilares: energia, edifícios, residências, transporte, tráfego, saúde e meio ambiente (GAUR, SCOTNEY, *et al.*, 2015).

Tecnologias voltadas à sustentabilidade também estão contidas nas cidades inteligentes (MAHALANK, MALAGUND, *et al.*, 2017), visto que estes sistemas têm o objetivo de diminuir os gastos/consumo de energia de uma cidade, além de otimização na produção de bens e serviços para a população.

Em relação à saúde, sistemas de software IoT surgem como alternativa para facilitar o acesso, por exemplo, a consultas e monitoramento de pacientes por meio da tecnologia. A cooperação de sensores e dispositivos torna mais fácil à identificação e

autenticação dos pacientes em diferentes aspectos. Para ATZORI, IERA, *et al.* (2010), há muitos benefícios em utilizar *healthcare*. A *healthcare* é agrupada em cinco categorias: (i) rastreamento de objetos e pessoas, (ii) identificação e autenticação, (iii) coleção de dados, (iv) sensoriamento e (v) ambientes inteligentes.

De modo geral, sistemas de software IoT possuem características multidisciplinares e inovadoras e estão inseridos em nosso cotidiano nas mais diversas áreas, como, por exemplo, na agricultura. Entre as vantagens de incorporar o paradigma IoT na agricultura podemos citar o mapeamento da fazenda, o monitoramento remoto das plantas, a verificação da distância e saúde dos animais, dentre outros. Como em outras áreas, a IoT se comporta da mesma maneira, com uso de técnicas artificiais para alterar fatores climáticos, como luz natural, temperatura e umidade, para criar condições ambientais adequadas para o crescimento de animais e plantas, além da utilização de atuadores, identificadores e sensores (GÓMEZ-CHABLA, REAL-AVILÉS, *et al.*, 2019, SHI, AN, *et al.*, 2019). Vale ressaltar que há diversas outras áreas nos quais o paradigma IoT atua e influencia, além das citadas nessa dissertação.

2.4.2. Características de Sistemas de software IoT

Esta subseção trata das principais propriedades funcionais e não funcionais que sistemas de software baseados em IoT deveriam apresentar. Para capturar estas propriedades, nesse trabalho foi utilizada a revisão realizada por Motta, Silva *et al.* (2019).

Motta, Silva *et al.* (2019) realizaram um estudo secundário com o propósito de caracterizar propriedades de sistemas de software baseados em IoT. Os autores encontraram e capturaram 29 características (Adaptabilidade, Autonomia, Conectividade, Disponibilidade, Eficiência, Escalabilidade, Heterogeneidade, Interoperabilidade, Privacidade, Segurança, Sensibilidade ao Contexto, dentre outros) e quatro comportamentos (**atuação, identificação, processamento e sensoriamento**) que estão relacionados diretamente a qualidade desse tipo de sistema.

Na pesquisa realizada por Ahmed, Bures, *et al.* (2019), os autores conduziram um mapeamento sistemático para identificar os principais aspectos de qualidade que devem ser considerados na construção de sistemas de software baseados em IoT. Em

um resultado similar ao encontrado por Motta, Silva *et al.* (2019), os autores identificaram que segurança, privacidade, desempenho, dentre outros aspectos de qualidade são essenciais na construção de sistemas de software IoT.

2.5 Considerações Finais

Este capítulo apresentou a revisão bibliográfica executada em busca dos conceitos dos tópicos principais relacionados a esta dissertação: (i) Inspeção de Software e (ii) Internet das Coisas.

O primeiro tópico apresentou as técnicas de inspeções de software existentes e voltadas para requisitos. Cada técnica foi descrita de modo objetivo, de modo a tornar mais claro a compreensão de cada uma. Também foram apresentadas as definições, taxonomias de defeitos e processo de aplicação de técnicas de inspeções em geral e que será utilizado nesta dissertação posteriormente.

A definição geral de Internet das Coisas que esta dissertação está levando em consideração o que foi apresentado no segundo tópico. As áreas de aplicação de IoT, bem como as propriedades que possui também foram descritas.

No próximo Capítulo serão apresentadas as técnicas de descrição de cenários IoT (*SCENARI_{IoT}*), as facetas IoT e a técnica proposta neste trabalho (*SCENARI_{IoT}CHECK*).

3 Proposta Inicial da Técnica **SCENAR_{IoT}CHECK**

Neste capítulo é apresentada a proposta inicial da técnica SCENAR_{IoT}CHECK – uma técnica de leitura baseada em checklist para verificação de cenários baseados em IoT. Esta técnica combina as perspectivas de duas abordagens: SCENAR_{IoT} e Facetas de IoT.

3.1 Introdução

Neste capítulo é apresentada a proposta inicial de uma tecnologia de verificação para apoiar a qualidade de artefatos de requisitos (cenários) baseados em sistemas IoT, chamada *SCENAR_{IoT}CHECK*. De acordo com Basili, Green, *et al.* (1996), o custo que se tem para corrigir os defeitos em artefatos de software aumenta exponencialmente com o decorrer do processo de desenvolvimento do sistema. Dessa forma, a *SCENAR_{IoT}CHECK* tem o objetivo de detectar defeitos nas descrições dos cenários IoT nas fases iniciais do ciclo de desenvolvimento de software.

Adicionalmente, esse capítulo apresenta os argumentos iniciais utilizados desde a criação até a finalização do *checklist*. A *SCENAR_{IoT}CHECK* combina duas abordagens que apoiam a construção de sistemas de software IoT. A primeira parte consiste na técnica *SCENAR_{IoT}* (SILVA, 2019), onde se trata de capturar características essenciais em sistemas de software IoT. A segunda parte consiste nas Facetas IoT (MOTTA, DE OLIVEIRA, *et al.*, 2018, MOTTA, DE OLIVEIRA, *et al.*, 2019), que trata da parte “não-funcional” do sistema.

3.2 **SCENAR_{IoT}**- uma técnica para especificação de cenários IoT

A *SCENAR_{IoT}* é uma técnica de especificação de cenários baseada em IoT (SILVA, 2019) que adapta e desenvolve tecnologias convencionais de cenários de software para apoiar a especificação de cenários IoT, considerando as características evidenciadas na literatura técnica. Essas características passaram a ser chamadas de arranjos de interação IoT (AII), como mostrado na Figura 4. Há um total de nove arranjos de interação IoT propostos por (SILVA, 2019).

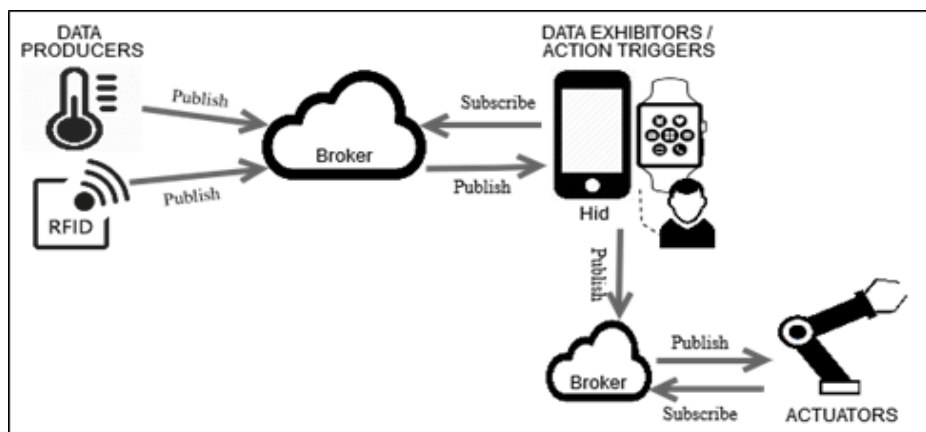


Figura 4. Exemplo de um All sobre atuação acionada por um indivíduo, com base nos dados da IoT (extraído de (SILVA, 2019))

Motta, Silva, *et al.* (2019) realizaram uma revisão estruturada para encontrar as principais características que os sistemas IoT possuem. Em seus resultados, os principais recursos e mecanismos de comportamento de IoT consideram as habilidades de **identificação**, **sensoriamento**, **processamento** e **atuação**. Cada um desses comportamentos permitiu a criação de arranjos de interação, que serão apresentados em seguida.

A **identificação** está relacionada a identificar as coisas do mundo real por meio de *tags*, *RFIDs* e outros. No **sensoriamento**, o objetivo é capturar dados por meio de sensores e hardwares imersos no ambiente. O **processamento** consiste em executar a análise e tomada de decisão dos dados. Finalmente, a **atuação** significa agir em uma determinada situação, tomando uma decisão sobre qualquer informação obtida, resultando em uma ação no ambiente. Esses três comportamentos são mapeados e vinculados aos sistemas IoT.

Uma vez identificadas essas características, Silva (2019) as combinou e propôs nove Arranjos de Interação IoT (Alls), representando um conjunto de características que devem ser capturadas quando engenheiros de software estão descrevendo cenários baseados em IoT. Esses arranjos orientam os engenheiros de software na captura de informações essenciais sobre o sistema que será construído. Além disso, cada All possui um catálogo contendo informações sobre todas as características a serem capturadas na descrição do cenário, conforme mostra a Figura 5. Todos os All (isolados ou combinados) resultam em um artefato de especificação de cenários para sistemas baseados em IoT.

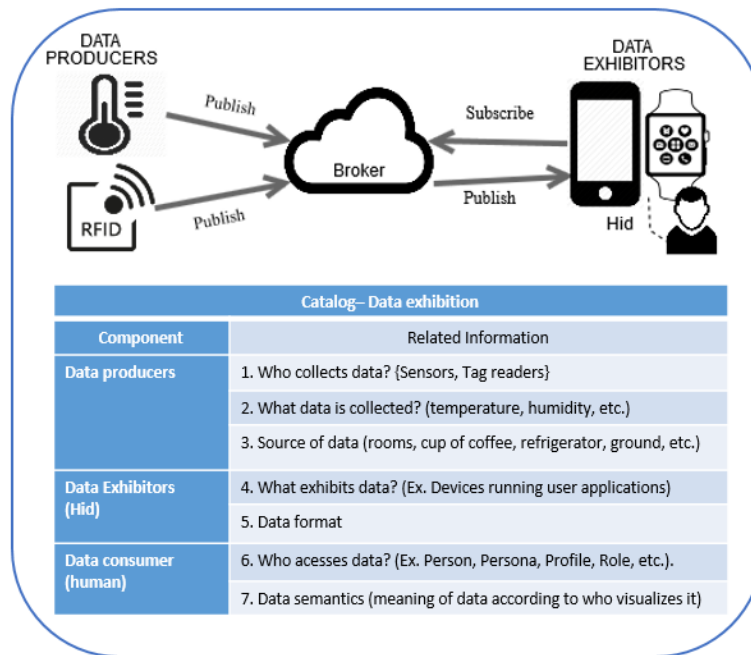


Figura 5. Exemplo de um All e seu catálogo associado (extraído de Silva (2019))

Assim, o artefato de descrição de cenário produzido pela *SCENARI_{IoT}* deve conter algumas das seguintes informações:

- 1) Identificação dos principais elementos dos sistemas de software IoT que irão compor a solução, bem como a forma como estes elementos são organizados;
- 2) Descrição do domínio principal do problema, como saúde, construções inteligentes, engenharia automotiva ou outras;
- 3) Descrição do papel de cada ator dentro do cenário (os atores podem ser usuários finais, coisas e sistemas de software);
- 4) Descrição das interações entre os atores e o sistema de software IoT;
- 5) Descrição do fluxo alternativo e de exceção no artefato;
- 6) Descrição da infraestrutura do ambiente IoT; e,
- 7) Descrição dos dispositivos que serão utilizados no sistema IoT.

No entanto, o trabalho de Silva (2019) possui algumas limitações no que diz respeito a descrição dos cenários: 1) a linguagem utilizada nesta técnica é de alto nível (linguagem do usuário); 2) não é proposto um *template* para que os engenheiros de software possam descrever os cenários IoT; 3) não garante a cobertura dos arranjos em capturar informações essenciais dos sistemas de software IoT para

descrição dos cenários; 4) embora tenha indicação de viabilidade e facilidade de uso, a técnica não foi ainda avaliada experimentalmente por profissionais atuantes na indústria; e, 5) mesmo a técnica sendo fácil de utilizar, ainda é necessário ter conhecimento sobre as características que os sistemas IoT possuem.

3.3 Facetas de IoT

No contexto de como projetar sistemas de software IoT e considerando suas características não-funcionais, Motta *et al.*, (2018) buscaram definir algumas características e preocupações que os profissionais e pesquisadores enfrentam ao lidar com a construção de tais sistemas. Para isso, Motta *et al.*, (2018) realizaram um estudo primário e secundário. No estudo secundário, eles buscaram características específicas de IoT. Enquanto no estudo primário, eles realizaram o *matching* das características do estudo secundário com o estudo primário obtido por meio da perspectiva dos profissionais.

Assim, Motta *et al.*, (2018) propuseram um *framework* contendo seis facetas e baseado no *framework* proposto por (ZACHMAN, 1987). As facetas levadas em consideração ao projetar sistemas de software IoT são: Conectividade, *Things*, Comportamento, *Smartness*, Interatividade e Ambiente (MOTTA, DE OLIVEIRA, *et al.*, 2019).

Cada faceta é organizada em uma matriz interpolando Interrogativas de Comunicação (colunas, 5W1H - o quê, como, onde, quem, quando e por quê) e Perspectivas (linhas) que especializam as facetas para desenvolver uma solução baseada em IoT considerando o 5W1H. A Figura 6 mostra a organização das facetas IoT; que é inspirado no *framework* de (ZACHMAN, 1987).

A definição de cada item da faceta permitiu a criação de perguntas específicas dentro do *checklist*. Em relação a **conectividade**, os autores definem que é necessário ter disponível um meio pelo qual as coisas podem se conectar. É essencial ter alguma forma de conexão como a internet. No entanto, os autores não limitam o tipo de conexão associando somente a internet. Eles também mencionam *bluetooth* e outros tipos de redes. **Comportamento** significa alguma ação que o sistema realizará dentro do ambiente por meio das coisas. O **Ambiente** está relacionado a inserção ou contexto em que o sistema estará imerso.

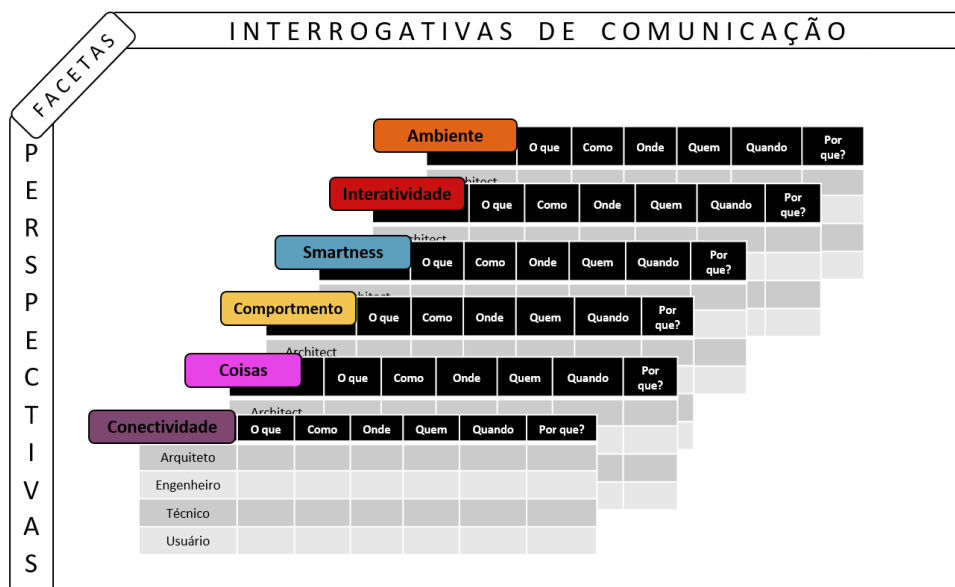


Figura 6. Framework para engenharia de sistemas baseados em IoT (extraído de (MOTTA, DE OLIVEIRA, *et al.*, 2019))

Com relação à **Interatividade** que ocorre dentro do sistema, Motta *et al.*, (2018) conceituaram que é o envolvimento de atores trocando informações com coisas e sistema. A **Inteligência** está relacionada ao comportamento do software, especificamente como um gerenciador para tomada de decisão. A Inteligência também se refere ao nível de inteligência que as coisas passam para o sistema de software. Em geral, um sistema de software IoT pode coletar dados do ambiente e usá-lo para tomar decisões e agir. **Things** indica as *coisas* em sistemas baseados em IoT, como *tags*, sensores, atuadores e todo o hardware que pode substituir o computador, expandindo o alcance da conectividade.

Com as definições de cada uma destas facetas, foi possível identificar algumas questões específicas para compor o *checklist* da técnica *SCENARI_{IoT}CHECK*.

3.4 Proposta Inicial da Técnica *SCENARI_{IoT}CHECK*

Esta subseção descreve o passo-a-passo da versão inicial da técnica *SCENARI_{IoT}CHECK*. Inicialmente, é demonstrado como foi realizada a junção dos All e Facetas IoT. Para isso, foi realizado um mapeamento das facetas dentro dos arranjos para se ter uma maior cobertura em relação as questões do *checklist* nas descrições dos cenários. Após isso, cada questão é descrita com suas respectivas propriedades.

3.4.1 Mapeamento das Facetas IoT nos AII

Para criação da técnica, primeiramente, foram identificadas e mapeadas as facetas IoT (MOTTA, DE OLIVEIRA, *et al.*, 2019) dentro dos AII (SILVA, 2019), buscando cobrir todas as informações que deveriam ser inspecionadas, como: atores, coisas (*things*), algoritmos empregados (*smartness*), tipos de interação e conexão utilizada nos cenários. Além disso, foram criadas perguntas para cobrir esse mapeamento.

Cada faceta está identificada por uma cor dentro dos arranjos. A faceta **things** é identificada em verde e representa o hardware no contexto dos sistemas IoT. Hardware significa sensores, atuadores, *wearables* ou qualquer outro dispositivo físico. A faceta de **conectividade**, identificada em vermelho, representa qualquer tipo de conexão que ocorre entre dispositivos, coisas e sistema de software IoT. Já a faceta **comportamental**, em amarelo, representa qualquer ação no contexto que o sistema de software IoT está inserido. A **interatividade** está representada pela cor azul, e está relacionada a todo tipo de interação existente entre os componentes no sistema de software IoT. O **ambiente**, representado pela cor roxa, é onde o sistema de software IoT está inserido. A **inteligência** ou "**smartness**" é representada pela cor azul claro.

- **Arranjo AII-1: Exibição de Dados**

Seu objetivo é capturar/obter dados do ambiente para o sistema principal por meio de um *broker* ou *gateway*. Em geral, ambiente, coisas, conexão e interação sempre estarão nos arranjos, dado que são características essenciais de sistemas de software IoT. Um exemplo deste sistema é a coleta de dados do ambiente de controle de poluentes, que captura os tipos de dados especificados e envia para um *dashboard* para os usuários visualizarem. De acordo com os autores, esse arranjo é o mais simples de ser identificado e, seu catálogo, preenchido. Como pode ser notado, este arranjo utiliza quatro das seis facetas. As perguntas feitas para esse arranjo são: *i) O tipo de dado coletado está especificado? ou; ii) É possível identificar quem ou o quê visualiza os dados?*

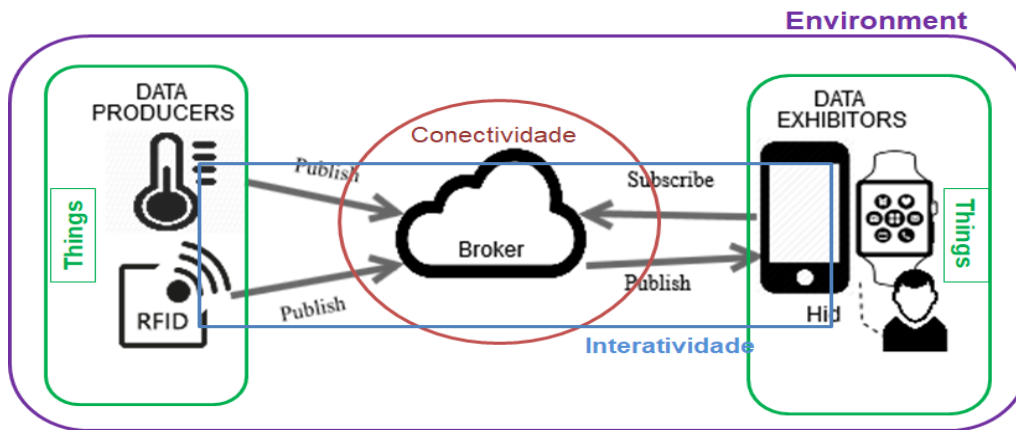


Figura 7. Coleta e exibição dos dados IoT para um indivíduo

- **Arranjo All-2: Atuação acionada por um indivíduo**

A parte de atuação (comportamento) em que se supõe que um usuário deseja ligar/desligar algo de maneira remota é trabalhada neste arranjo. Nota-se que quem toma a decisão sobre determinada informação é o próprio usuário final. *Things* está relacionada aos *wearables* e atuadores que o sistema possuirá. As perguntas feitas por esse arranjo são: i) *É possível identificar quem ou o que visualiza os dados?* Ou ii) *É possível identificar quem ou o que realiza a ação ou atua no sistema?*

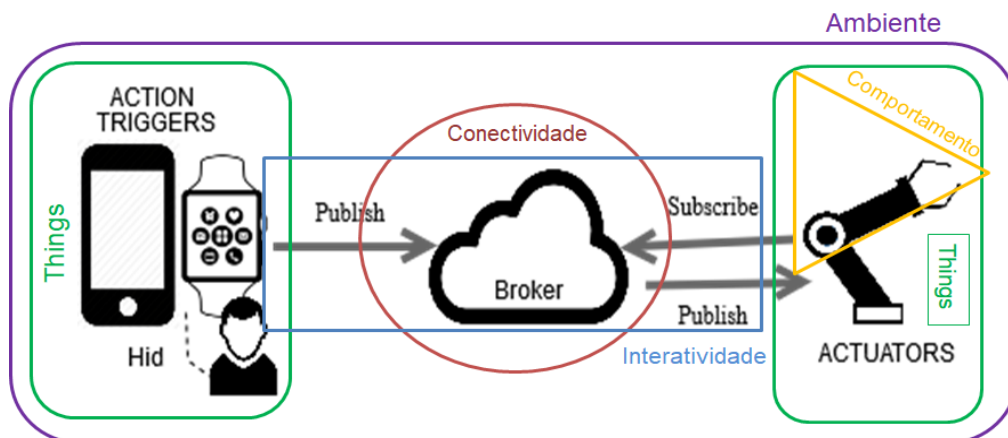


Figura 8. Indivíduo acionando uma atuação no ambiente

- **Arranjo All-3: Atuação acionada por um sistema de software**

Este arranjo é similar o All-2, diferenciando-se apenas no ator que tomará a decisão. Diferente All-2, em que o usuário toma a decisão, neste arranjo, o sistema de software atua como tomador decisão. A parte do protocolo de comunicação também é o *broker* ou *gateway*. Assume-se que o sistema já

possui alguns dados coletados, sobre uma determinada situação, para tomar decisão (autonomamente) e depois atuar sem intervenção humana. As perguntas feitas por esse arranjo são: i) *O objetivo do sistema está descrito detalhadamente? (Somente visualização de dados; visualização, tomada de decisão e atuação).* ii) *É possível identificar quem ou o que realiza a ação ou atua no sistema?*

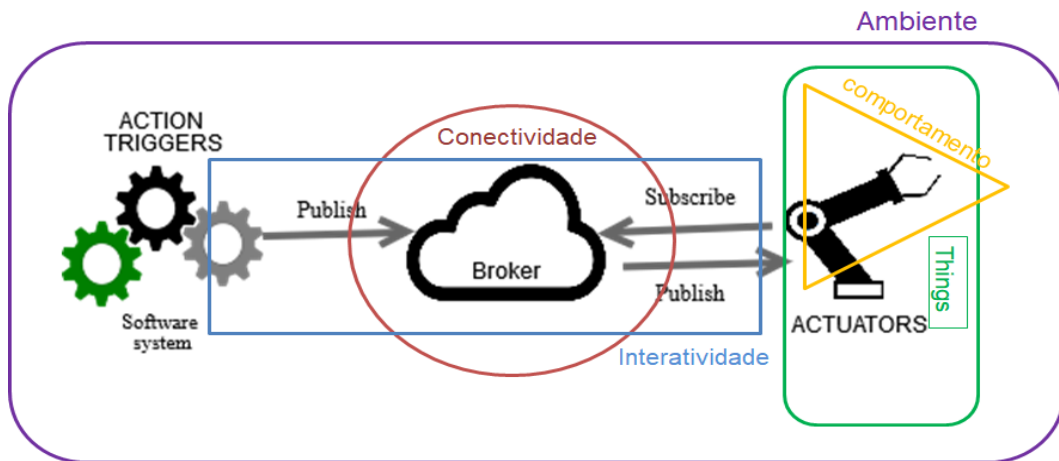


Figura 9. Sistema de software acionando uma atuação no ambiente

- **Arranjo All-4: Atuação acionada por um indivíduo, com base nos dados da IoT**

Este arranjo é uma combinação de dois outros arranjos (All-1 e All-2), em que os sensores coletam dados do ambiente e enviam para um usuário final (no caso, humano) para que ele tome uma decisão sobre a informação recebida. E posteriormente, o sistema atua no ambiente realizando alguma atividade/comportamento. As perguntas feitas para esse arranjo são: i) *Os eventos que o sistema possui foram identificados? (ex: ligar/desligar um objeto, enviar dados) ou ii) É possível identificar quem ou o que gerencia os dados coletados? (Administrador, decision maker, usuários).*

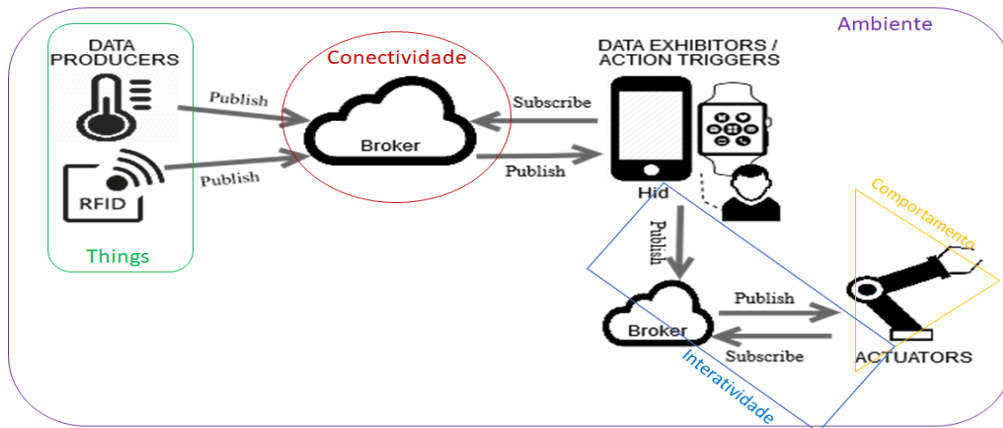


Figura 10. Indivíduo acionando uma atuação com base em dados da IoT

- **Arranjo All-5: Atuação acionada por um sistema de software, com base em dados da IoT**

Este arranjo é uma combinação de dois outros arranjos (All-1 e All-3), em que os sensores coletam dados do ambiente e enviam para um sistema de software para que ele tome uma decisão sobre a informação recebida. E posteriormente, o sistema atua no ambiente realizando alguma atividade/comportamento. Vale ressaltar que este é o primeiro arranjo que possui as seis facetas mapeadas dentro dele, dando a característica de “total autonomia”. As perguntas feitas para esse arranjo são: i) *Os eventos que o sistema possui foram identificados? (ex.: ligar/desligar um objeto, enviar dados) ou ii) É possível identificar quem ou o que gerencia os dados coletados? (Administrador, decision maker, usuários).*

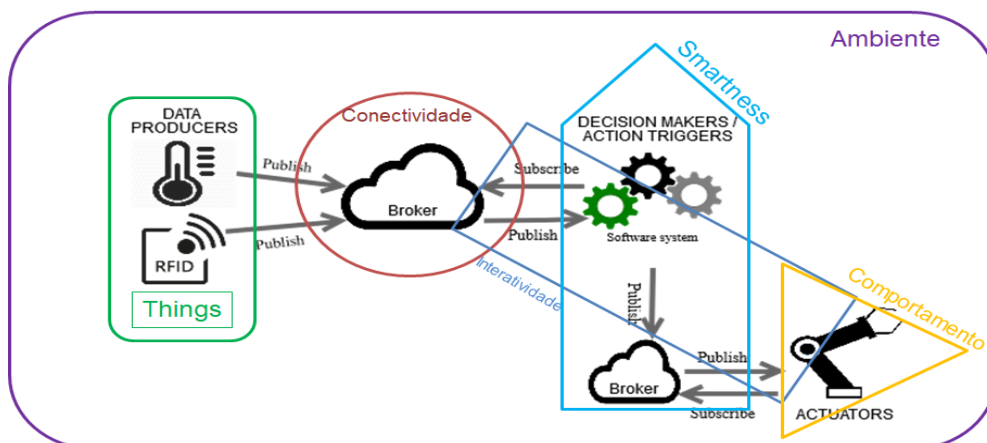


Figura 11. Sistema de software acionando uma atuação com base em dados IoT

- **Arranjo All-6: Atuação acionada por um sistema de software, com base em dados não-IoT**

Neste arranjo, as fontes dos dados são provenientes de outros sistemas de software, como um sistema de *previsão do tempo* ou de uma *rede social* como *Twitter*. Estes dados são classificados como “não-IoT” devido ao fato de não serem coletados diretamente por sensores. Este tipo de arranjo também trabalha com autonomia, sem necessidade de um humano para tomar decisão. As perguntas feitas para esse arranjo são: i) *É possível identificar a fonte de provimento dos dados?* ou ii) *É possível identificar quem ou o que gerencia os dados coletados?* (Administrador, decision maker, usuários).

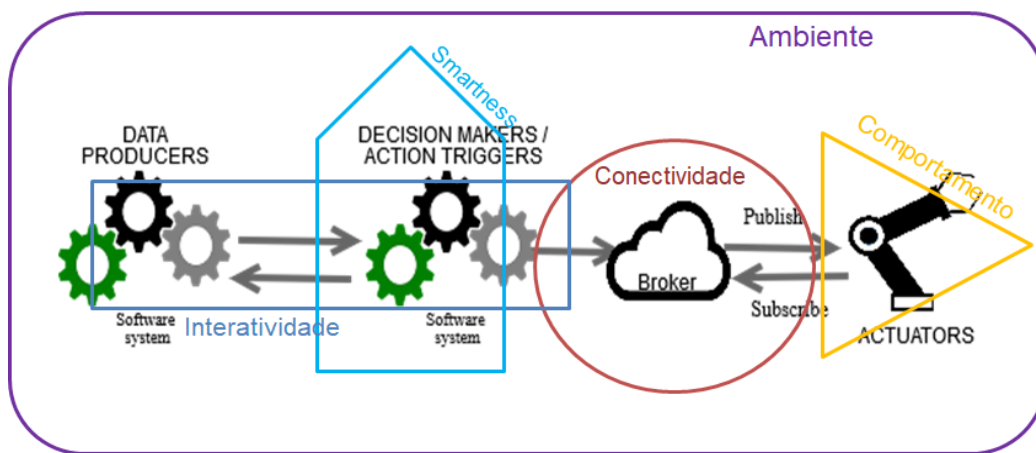


Figura 12. Sistema de software acionando uma atuação com base em dados não IoT

- **Arranjo All-7: Atuação não-IoT acionada por um sistema de software, com base em dados da IoT**

Neste arranjo, as fontes dos dados são provenientes de sensores, ou seja, são dados IoT. Ele também possui a característica de *Smartness*, dando autonomia para o sistema baseado em IoT. No entanto, sua atuação pode ser exemplificada como “o envio de uma notificação” ou “postar algo em alguma rede social”. As perguntas feitas para esse arranjo são: i) *É possível identificar quem ou o que coleta os dados?* Ou ii) *É possível identificar como o sistema reage de acordo com as alterações ocorridas no ambiente?*

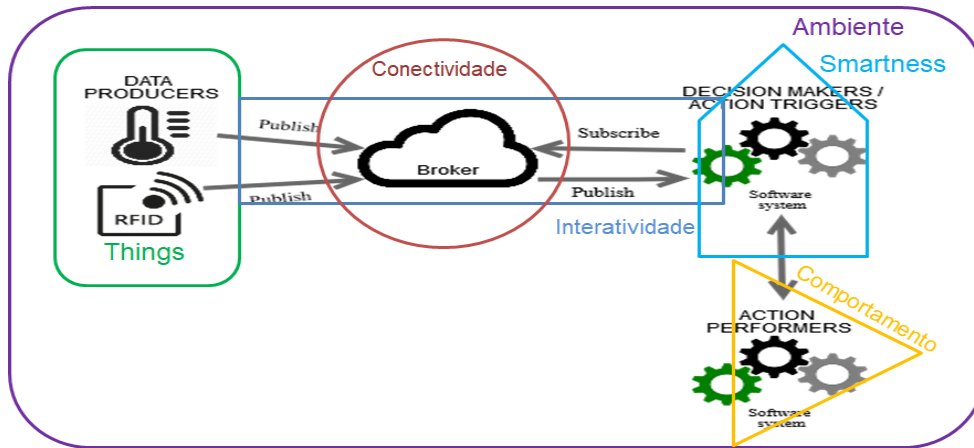


Figura 13. Sistema de software acionando uma atuação não IoT com base em dados IoT

- **Arranjo All-8: Atuação acionada por um indivíduo, com base em dados não-IoT**

Este arranjo trabalha com dados não-IoT para tomada de decisão por um usuário e possui cinco das seis facetas mapeadas dentro dele. Os produtores de dados são outros sistemas de software como sistemas de previsão do tempo ou rede social. Os dados obtidos são passados para um usuário final (humano) que toma uma decisão e o sistema precisa atuar no ambiente em que está imerso. As perguntas feitas para esse arranjo são: *i) É possível identificar a fonte de provimento de dados? Ou ii) É possível identificar quem visualiza os dados? (Coisas, sistemas de software, usuários).*

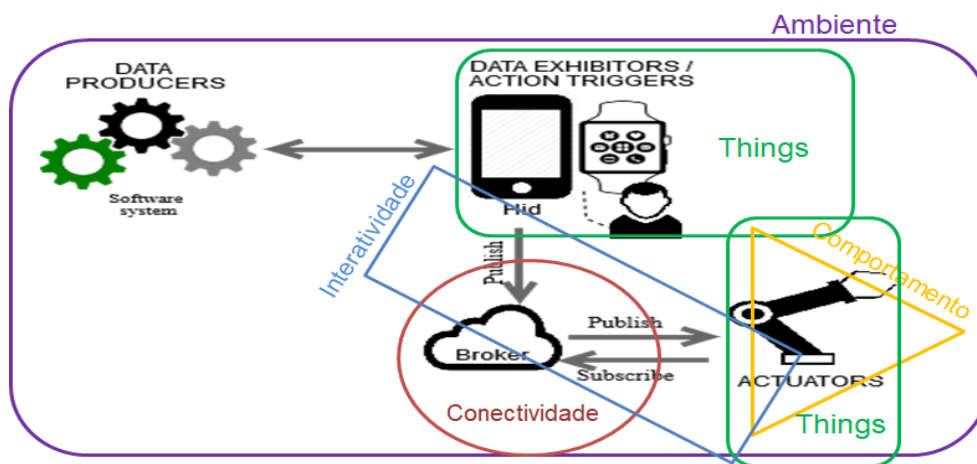


Figura 14. Indivíduo acionando uma atuação com base em dados não IoT

- **Arranjo All-9: Atuação não-IoT acionada por um indivíduo, com base em dados da IoT**

Este arranjo trabalha com dados IoT obtidos de sensores para tomada de decisão por um humano. No entanto, o tipo de atuação é não-IoT, devido ao usuário simplesmente “enviar uma notificação” ou realizar um “post” em uma rede social baseado nos dados obtidos pelos sensores. Em geral, as perguntas feitas para este arranjo são: *i) É possível identificar a fonte de provimento de dados? Ou ii) É possível identificar quem ou o que acessa os dados coletados? (Coisas, sistemas de software, usuários).*

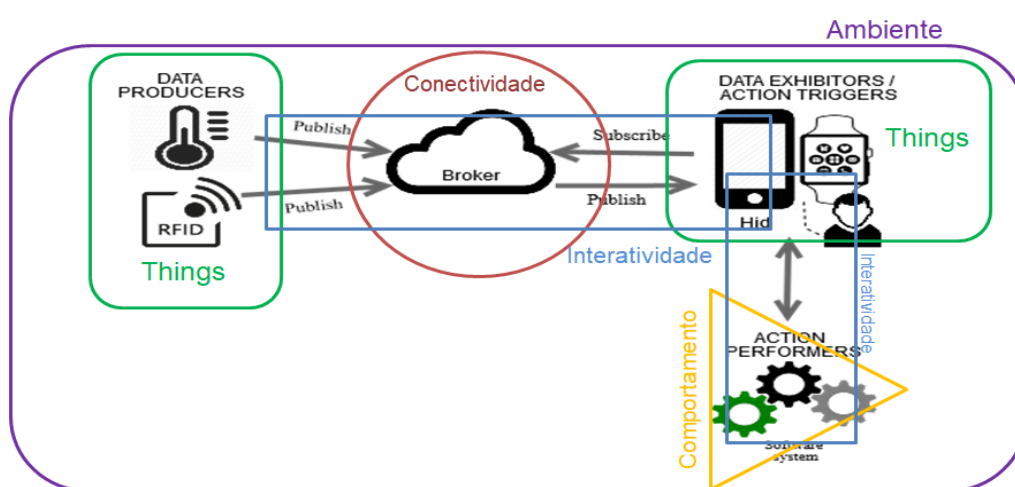


Figura 15. Indivíduo acionando uma atuação não IoT com base em dados IoT

3.4.2 Técnica *SCENARI_{IoT}CHECK*

Nesta subseção, é apresentada a técnica *SCENARI_{IoT}CHECK* e a caracterização das suas respectivas questões do *Checklist*. Com o mapeamento das facetras dentro dos arranjos, foi possível elaborar as questões do *checklist* de modo a cobrir as principais características desses sistemas. As características que estamos levando em consideração estão disponíveis na revisão estruturada realizada por (MOTTA, SILVA, *et al.*, 2019).

Cada questão do *checklist* procura identificar se as informações necessárias foram capturadas e descritas nos cenários baseados em IoT. Se essas informações não foram capturadas, isso significa que existe um possível problema (uma discrepância) no artefato inspecionado. Além disso, é descrito o raciocínio para cada questão da lista de verificação. As questões 1 a 13 referem-se aos Alls proposto por

(SILVA, 2019). A Tabela 5 mostra a caracterização de uma questão do *checklist* e como a associação foi realizada entre as questões e os AIs. O restante da caracterização das questões está disponível no Apêndice D.

Tabela 5. Exemplo de caracterização das questões providas dos arranjos

Item de Avaliação - Questão 1	
Descrição do Item	O objetivo do sistema está descrito detalhadamente? (Somente visualização de dados; visualização, tomada de decisão e atuação)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar a finalidade para qual o sistema será construído
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### AIII</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados.

As questões 14 a 25 referem-se as facetas com questões específicas dentro do contexto de sistemas baseados em IoT (MOTTA, DE OLIVEIRA, *et al.*, 2019). A Tabela 6 mostra a caracterização de uma questão do *checklist* para cobrir características encontradas pelas facetas.

Tabela 6. Exemplo de caracterização das questões providas das Facetas IoT

Item de Avaliação - Questão 14	
Descrição do Item	É possível identificar o contexto no qual o sistema está inserido? (cidade inteligente, agricultura, automobilística, assistência em saúde)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	O objetivo desta questão é verificar em qual contexto a aplicação será inserida. É essencial possuir esta informação devido a características distintas de cada domínio do problema
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### AIII</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	Um sistema para controle da poluição de uma parte da cidade do Rio de Janeiro. Após uma certa quantidade de dados coletados. O sistema analisa esses dados e toma uma decisão. Tomar uma decisão não implica em gerar uma ação no ambiente.

Com relação ao processo de uso da *SCENARI_{OT}CHECK*, as descrições dos cenários são produzidas utilizando a técnica de especificação *SCENARI_{OT}*. Em seguida, os inspetores aplicam a técnica *SCENARI_{OT}CHECK*, detectando possíveis defeitos nos cenários. Por fim, é produzido um novo artefato com as correções, após a reunião de discriminação, contendo a nova descrição do cenário. A Figura 16 mostra como as duas técnicas devem ser aplicadas.

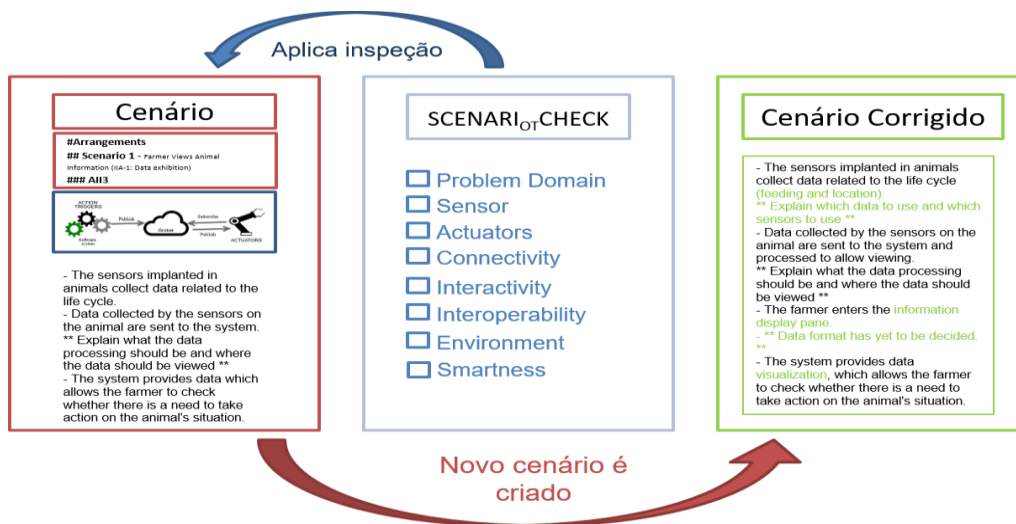


Figura 16. Processo de aplicação das duas técnicas *SCENARI_{OT}* e *SCENARI_{OT}CHECK*

3.5 Considerações Finais sobre o Capítulo

Este capítulo apresentou a técnica *SCENARI_{OT}CHECK* (a *Checklist for Reading IoT-based Scenarios*), uma técnica de inspeção proposta para avaliar cenários de sistemas baseados em IoT. Esta técnica combina duas perspectivas das abordagens *SCENARI_{OT}* e Facetas de IoT. Assim como foi apresentado o *rationale* de sua criação.

A primeira parte da *SCENARI_{OT}* busca inspecionar todos os elementos de um modo geral nos cenários. Enquanto a segunda parte da técnica, que trata das facetas, busca cobrir e inspecionar conteúdo específico nos cenários descritos.

O próximo capítulo descreve um estudo de viabilidade executado inicialmente, de acordo com a metodologia utilizada neste trabalho e baseada em experimentação para a avaliação e aprimoramento da técnica *SCENARI_{OT}CHECK*. A primeira versão do *checklist* está disponível no Apêndice B em que ela foi utilizada para o primeiro estudo de viabilidade.

4 Avaliando a Primeira Versão da SCENARI_{IoT}CHECK

Este capítulo descreve um estudo de viabilidade executado de acordo com a metodologia proposta por (Shull, Carver, 2001) que direcionou a avaliação da técnica SCENARI_{IoT}CHECK. Os resultados desse estudo forneceram indícios de sua viabilidade.

4.1 Introdução

A engenharia de software utiliza técnicas e/ou evoluções de práticas de observação de outras ciências para experimentalmente avaliar suas abordagens, processos, tecnologias ou métodos. Portanto, neste capítulo, é apresentado um estudo de viabilidade para verificar a SCENARI_{IoT}CHECK (SC) no contexto de sua utilização para a inspeção de cenários baseados em IoT. Conforme a metodologia que esse trabalho está seguindo, o estudo de viabilidade tem o objetivo de verificar se a abordagem proposta consegue alcançar seu objetivo.

4.2 Estudo de Viabilidade

De acordo com a metodologia proposta por Shull, Carver e Travassos (2001), o estudo de viabilidade é o primeiro estudo a ser conduzido para avaliar uma tecnologia recém-criada e verificar sua viabilidade mediante o tempo e/ou objetivo proposto.

A técnica SCENARI_{IoT}CHECK, objeto dessa dissertação, foi observada, inicialmente, em um estudo de viabilidade em um minicurso de graduação, cujos participantes apresentavam grau de experiência e conhecimento similares em relação a sistemas de software IoT. Os participantes foram selecionados por conveniência, dado que se tratava de um estudo inicial para verificar se a técnica seria viável ou não. O restante dessa seção está dividido em planejamento, execução, resultados e ameaça a validade do estudo.

4.2.1 Planejamento

Nesta etapa, o design do estudo foi preparado com todos os artefatos produzidos e que foram utilizados pelos participantes e pesquisadores (Apêndice A). Foi elaborado um Termo de Consentimento, um formulário de caracterização dos

participantes, além do problema a ser tratado no estudo juntamente com seus respectivos cenários. Para garantir consistência e adequação, foi realizado um estudo piloto para observar a qualidade dos instrumentos planejados e construídos.

O objeto de estudo é a técnica de inspeção (*SCENARI_{IoT}CHECK*) para inspecionar a descrição dos cenários baseados em IoT e sua intenção é observar a viabilidade de aplicação e discutir os resultados de seu primeiro estudo de viabilidade. Assim, a principal questão deste estudo é:

“A técnica SCENARI_{IoT}CHECK apoia a identificação de defeitos nas descrições de cenários para sistemas de software IoT?”

Em seguida, o GQM (BASILI, CALDIERA, *et al.*, 1994) foi utilizado para alinhar a questão de pesquisa com o objetivo e as métricas utilizadas na pesquisa. O estudo experimental tem o objetivo de **analisar** a técnica *SCENARI_{IoT}CHECK* **com o propósito de** caracterizar **em relação à** sua viabilidade, observada em termos de custo-eficiência, eficiência e eficácia **do ponto de vista** dos engenheiros de software **no contexto de** desenvolvedores (representados pelos participantes de um curso de verão sobre requisitos de software) avaliando um artefato de descrição de cenário da IoT produzido com a técnica de especificação *SCENARI_{IoT}*.

O estudo foi realizado durante um curso de verão em uma universidade do norte do Brasil no início de 2019. Durante o curso, os tópicos discutidos foram: i) sistemas de software IoT: definições, exemplos e suas aplicações; e ii) especificações e inspeções de cenários para sistemas de software IoT, bem como suas aplicações.

As variáveis dependentes são: custo-eficiência⁵, eficiência⁶, eficácia⁷ e número de defeitos. Em relação às variáveis independentes, são considerados o tempo e a experiência dos participantes. Não foi possível colocar alguma hipótese a prova, uma vez que o número de participantes representou uma amostra de tamanho reduzido, inviabilizando a aplicação de testes estatísticos.

⁵ Custo-eficiência: razão entre os defeitos detectados por cada grupo dividido pelo tempo de cada grupo.

⁶ Eficiência: razão entre defeitos detectados por cada grupo dividido pelo número de discrepâncias.

⁷ Eficácia: razão entre defeitos detectados por cada grupo dividido pelo número total de defeitos conhecidos no artefato inspecionado.

O ambiente acadêmico foi selecionado nesse primeiro estudo para observar a técnica como indicado em (SHULL, CARVER, 2001). Segundo Barcelos (2006), é necessário realizar estudos experimentais em ambientes acadêmicos, como um passo inicial, para reduzir o risco de transferência de tecnologias “imaturas” para o setor produtivo privado.

4.2.2 Execução

4.2.2.1 Estudo Piloto

Dois membros (P1 e P2) do grupo de Engenharia de Software Experimental da COPPE/UFRJ participaram do estudo piloto. Ambos os participantes declararam possuir experiência relevante em IoT e desenvolvimento de software. O objetivo do estudo piloto foi verificar a qualidade dos materiais e procedimentos antes de aplicá-lo no estudo de viabilidade. Após uma breve introdução ao material de apoio, os participantes foram apresentados a dois cenários baseados em IoT, onde deveriam descrevê-los utilizando os AIs da técnica *SCENARI_{IoT}* (SILVA, 2019) juntamente com as Facetas IoT (MOTTA, DE OLIVEIRA, *et al.*, 2019). Em seguida, foi realizado o *cross-checking* dos cenários produzidos pelos participantes e foi solicitado a eles que realizassem a inspeção dos artefatos usando a *SCENARI_{IoT}CHECK*. O P1 inspecionou a descrição do cenário preparado pelo inspetor P2 e vice-versa, seguindo o arranjo do estudo. Como resultado, os participantes retornaram à descrição dos cenários além do documento inspecionado, os quais foram posteriormente avaliados e evoluídos de acordo com sugestões deles para que fosse possível a aplicação do estudo de viabilidade com os artefatos corrigidos e sem ambiguidade.

4.2.2.2 Execução

No estudo de viabilidade, 16 participantes foram divididos em quatro grupos (G1, G2, G3 e G4). Doze participantes eram alunos dos cursos de graduação em engenharia de software e sistemas de informação. Quatro participantes eram professores (dois com grau de doutor e dois com grau de mestre) da própria instituição. Todos os participantes assinaram um termo de consentimento antes da execução do estudo e, em seguida, um formulário de caracterização. A execução e análise dos resultados referentes aos dados coletados nesse estudo são apresentados nas próximas subseções.

Todos os participantes participaram de treinamentos sobre a descrição e inspeção dos requisitos baseados em IoT. O treinamento foi ministrado durante quatro dias, com duração de duas horas cada sessão. Alguns problemas do mundo real foram contextualizados nas tarefas para que os participantes tivessem mais experiência na especificação de cenários baseados em IoT. Também foi garantido que todos os grupos tivessem o mesmo treinamento e realizassem atividades semelhantes simultaneamente.

O problema proposto se relacionou a construção de um sistema de software para uma “estufa inteligente”. A ideia principal foi construir um sistema para uma estufa inteligente, onde deveria ser possível mudar seu comportamento de estufa convencional para semi-automatizada, utilizando o paradigma de sistemas de software IoT. Os participantes foram solicitados a descrever os cenários propostos em relação a uma solução para esse sistema. Na avaliação dos resultados, foram analisados os cenários baseados em IoT resultantes para descobrir eventuais problemas de requisitos.

Inicialmente, o conceito de IoT foi apresentado e discutido. Com relação aos conceitos iniciais, as atividades e problemas foram apresentadas a serem resolvidas em sala de aula, além de tarefas extras. Após isso, foram apresentadas, respectivamente, as técnicas de especificação e inspeção de cenários IoT – *SCENARI_{IoT}* e *SCENARI_{IoT}CHECK* (Apêndice B).

A primeira técnica (*SCENARI_{IoT}*) foi utilizada para descrever os cenários por cada grupo. Após a descrição dos cenários, foi dado um treinamento sobre inspeção de software. E posteriormente, os participantes realizaram a inspeção de algum cenário. A tarefa de cada grupo foi descrever dois cenários baseados em IoT utilizando a técnica de especificação *SCENARI_{IoT}*. Cada grupo gerou duas descrições de cenário, totalizando oito diferentes cenários baseados em IoT a serem inspecionados. No primeiro momento, foi realizada a inspeção *ad-hoc* de acordo com o planejamento do estudo. Dessa forma, os cenários produzidos pelo G1 foram dados ao G2 para realizar uma inspeção *ad-hoc* e vice-versa. Vale ressaltar que os grupos não tiveram nenhum tipo de auxílio, além do material apresentado previamente em sala de aula e durante o treinamento.

Em seguida, o último treinamento foi ministrado para os grupos com a técnica *SCENARI_{IoT}CHECK*. Nesta última tarefa, os cenários produzidos pelo G1 foram dados ao G3 para realizar a inspeção e vice-versa (trocados entre todos os grupos).

Portanto, todos os grupos tiveram a chance de utilizar a primeira técnica inspeção *ad-hoc* e, em seguida, a técnica *SCENARI_{IoT}CHECK*. É essencial dizer que os grupos tiveram acesso aos cenários baseados em IoT propostos por um outro grupo somente no momento em que inspecionaram os artefatos.

4.2.2.3 Resultados da Caracterização

Com o formulário de caracterização, foi possível obter uma visão geral da experiência e conhecimento que cada grupo possuía em relação a especificação e inspeção de requisitos, além de sistemas de software IoT (descrito na Tabela 7).

Tabela 7. Caracterização de cada grupo

Grupo	Conhecimento em Descrição de Cenários		Conhecimento de Inspeção de Software	Conhecimento dos sistemas baseados em IoT
G1	Baixo e Médio	Baixo	Baixo	
G2	Baixo	Baixo e Médio	Nenhum e Baixo	
G3	Médio	Baixo	Baixo	
G4	Nenhum, Baixo, Médio e Alto	Nenhum, Baixo e Médio	Nenhum e Baixo	

4.2.3 Resultados Qualitativos

Com os resultados qualitativos, foi possível observar como os grupos aplicaram a técnica. Portanto, após a aplicação da técnica *SCENARI_{IoT}CHECK*, os participantes foram convidados a responder um questionário final sobre sua utilidade e compreensão.

A apresentação da análise dos dados qualitativos sobre a percepção dos participantes para identificar as dificuldades enfrentadas por eles em relação à compreensão e utilidade das questões da *SCENARI_{IoT}CHECK*.

A análise dos resultados qualitativos foi realizada de forma *ad-hoc*. Os dados sobre compreensão e utilidade foram obtidos por meio de um questionário *pós-inspeção*. O questionário sobre pós-inspeção está disponível no Apêndice E.

Uma pergunta em relação a compreensão da técnica *SCENARI_{IoT}CHECK* seria: “Na sua opinião, qual aspecto da técnica torna a sua aplicação fácil/difícil de utilizar?” Seguem algumas das respostas dos participantes:

“Perguntas objetivas e diretas auxiliam na análise do cenário avaliado.”

Participante do grupo 1.

“A simplicidade das perguntas.” - Participante do grupo 1.

“Foi fácil porque tudo é claro e objetivo. Acho que seria mais fácil se eu tivesse mais conhecimento sobre a inspeção.” - Participante do grupo 3.

“Por ter perguntas fáceis de entender, a técnica se torna fácil.” - Participante do grupo 3.

“O checklist torna o processo fácil.” - Participante do grupo 4.

Em geral, os participantes relataram que as perguntas do *checklist* eram claras e utilizavam linguagem de fácil compreensão. Também foram feitas algumas perguntas fechadas, como mostra a Figura 17, na qual onze participantes concordam que o *SCENARI_{IoT}CHECK* é uma técnica fácil de aprender.

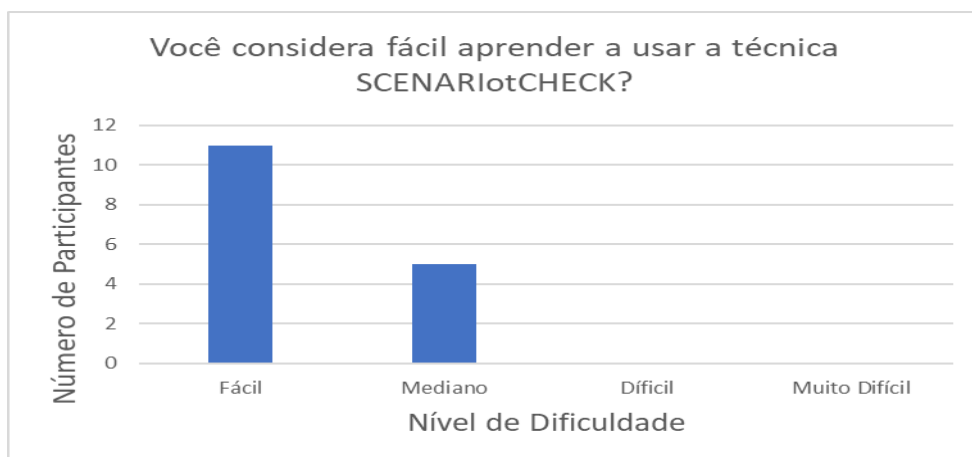


Figura 17. Compreensão da *SCENARI_{IoT}CHECK*

Quanto à utilidade, os participantes responderam à seguinte pergunta: “Você considera que as questões da técnica *SCENARI_{IoT}CHECK* inspecionam todos os elementos dos cenários descritos? Se “sim” ou “não”, por quê?”

Nesta questão, entendeu-se que a técnica de *SCENARI_{IoT}CHECK* foi útil para os participantes, uma vez que não conheciam alguns aspectos dos sistemas de software IoT e pela falta de experiência com tais sistemas. Além disso, pelo menos um participante de cada grupo relatou que a técnica *SCENARI_{IoT}CHECK* ajudou a detectar defeitos nos artefatos mais do que a inspeção *ad-hoc*.

“Até agora não perdi nenhum item avaliado em uma inspeção.” - Participante do grupo 1.

“Sim, porque avalia desde o aspecto de interação até a infraestrutura do cenário.” - Participante do grupo 2.

“Sim, porque com a ferramenta, podemos identificar todos os elementos do cenário.” - Participante do grupo 4.

“Sim, porque identifica a parte sobre a inspeção dos elementos.” - Participante do grupo 3.

“Sim, porque é possível entender e identificar as informações corretamente.” - Participante do grupo 1.

Sobre a questão fechada em relação a utilidade, foi observado que a técnica SCENARI_{IoT}CHECK ajudava os participantes a detectar defeitos na descrição de cenários IoT. Os resultados mostram que todos os participantes concordaram que a técnica os apoiou a identificar defeitos nos cenários IoT. A Figura 18 mostra o grau de utilidade da técnica de acordo com o *feedback* dos participantes.

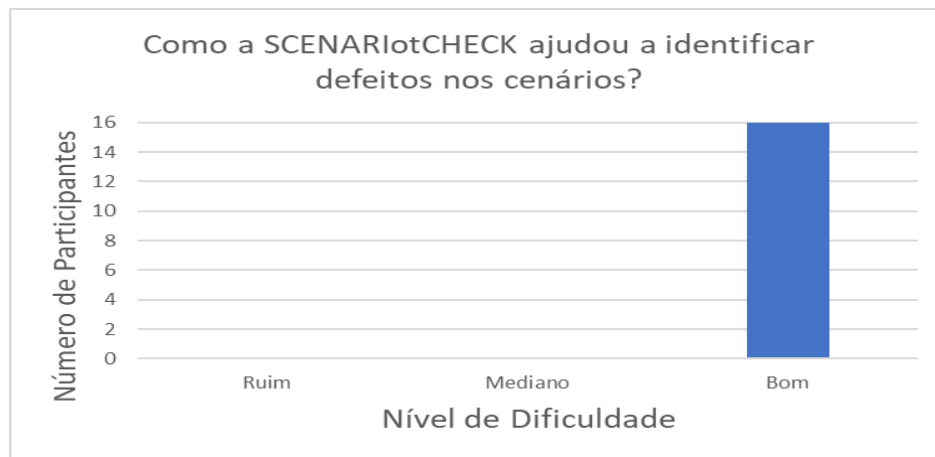


Figura 18. Utilidade da SCENARI_{IoT}CHECK

Resultados adicionais também foram coletados, alguns *feedbacks* para melhoria da técnica foram obtidos, o que ajudou a identificar questões ambíguas e a melhorá-las. Alguns dos comentários foram:

“Não, porque pode ocorrer a partir de elementos que aparecem ou passando elementos que não foram notados.” - Participante do grupo 1.

“Não, porque a técnica é mais superficial (sem entrar em termos técnicos).” - Participante do grupo 2.

Em aspectos gerais da técnica de SCENARI_{IoT}CHECK, foi solicitado aos participantes que dessem seus comentários sobre sua experiência no uso da técnica.

“A técnica tem informações (perguntas) que considero as mesmas. Ex: questão 2 = questão 14” - Participante do grupo 4.

“A pergunta 13 não está clara” - Participante do grupo 4.

“Algumas perguntas podem ficar mais claras, como as perguntas 08 e 11” -
Participante do grupo 1.

“Eu colocaria mais um item de inspeção (dados); por exemplo, é possível identificar quais dados devem ser inseridos / coletados?” - Participante do grupo 4.

4.2.4 Resultados Quantitativos

Para investigar a eficácia, custo-eficiência e eficiência da *SCENARIOTCHECK* (SC), foi realizada uma comparação com a eficácia, custo-eficiência e eficiência da técnica *ad-hoc* (AH). O objetivo da comparação não foi para observar qual técnica era mais eficiente ou eficaz. A técnica AH foi utilizada como um oráculo para verificar se a técnica SC alcançava seu objetivo.

Após a etapa da reunião de discriminação, para investigar a eficácia do SC (EFC), eficiência (EFF) e custo-eficiência (CE), os resultados foram comparados a técnica AH. Os grupos identificaram 45 defeitos no conjunto de cenários de IoT usando as técnicas AH e SC com a remoção das duplicatas e falso-positivos.

Na avaliação da técnica SC, os grupos identificaram um total de 36 defeitos no tempo médio de 0,6 horas; enquanto a inspeção utilizando a técnica AH, os grupos detectaram nove defeitos em um tempo médio de 0,924 horas nos cenários IoT avaliados.

Devido ao tamanho da amostra e às características de agrupamento, somente estatística descritiva foi utilizada. Além disso, a média e mediana de ambas as técnicas foram calculadas. Quanto à relação de **custo-eficiência**, os resultados da técnica de SC foram: a média foi de 16,569; e a mediana foi 14,141. Sobre a técnica de AH, os resultados foram: a média foi de 2,4; a mediana foi de 2,719. Pode-se notar que o CE do SC é maior que o CE da AH.

Em relação à **eficácia**, os resultados quantitativos da técnica SC foram: a média foi de 0,875; e a mediana foi de 0,916. Sobre a técnica de AH, os resultados foram: a média foi de 0,276; e a mediana foi de 0,297. Pode notar também que a eficácia da técnica SC é maior do que a eficácia da inspeção AH. Os resultados da **eficiência** da técnica SC foram: a média foi de 0,615; e a mediana foi de 0,649. Em relação a técnica de AH, os resultados foram: a média foi de 0,408; e a mediana foi de 0,386.

Os resultados por cada grupo são mostrados na Tabela 8, onde é apresentada a comparação da técnica AH e os resultados do SC com relação ao tempo, custo-eficiência, efetividade, discrepância, defeitos falso-positivos e reais.

Tabela 8. Resumo dos resultados obtidos no estudo.

G	Tempo (h)	AH disc	AH FP	AH DR	AH CF	AH EFF	AH EFC	Tempo (h)	SC disc	SC FP	SC DR	SC CE	SC EFF	SC EFC
G1	0,867	7	3	4	4,614	0,571	0,364	0,733	7	2	5	6,821	0,714	1
G2	0,933	5	4	1	1,072	0,2	0,111	0,667	19	7	12	17,991	0,631	0,923
G3	1,23	4	1	3	2,439	0,75	0,231	0,583	9	3	6	10,292	0,667	0,667
G4	0,667	9	8	1	1,5	0,111	0,4	0,417	29	16	13	31,175	0,45	0,909
Média	0,924	6,25	4	2,25	2,4	0,408	0,276	0,6	16	6,75	9	16,569	0,615	0,875

Legenda: FP: falso positivo; DR: defeitos reais; G: Grupo; EFC: eficácia; EFF: eficiência; CE: custo-eficiência

O resumo da média dos resultados é mostrado na Tabela 9, onde é apresentada a comparação dos resultados da inspeção AH e da técnica SC com relação ao tempo, defeitos detectados (todos os grupos), indicadores de custo-eficiência, eficiência e eficácia.

Tabela 9. Resumo dos indicadores das médias

Técnica	Média Defeitos Detectados	Média do Tempo (H)	Indicador Médio de Custo-eficiência (%)	Indicador Médio de eficiência (%)	Indicador Médio de Eficácia (%)
AH	2,25	0,924	2,4	0,408	0,276
SC	9	0,6	16,569	0,615	0,875

Os resultados obtidos forneceram indicação da viabilidade do uso da técnica SC para apoiar a inspeção de especificações de cenários baseados em IoT produzidas com o *SCENARI_{IoT}* e detectar defeitos. Com esses resultados, a técnica SC evoluiu para uma segunda versão, incrementada com mais perguntas.

Em relação à custo-eficiência, eficiência e eficácia, de acordo com resultados, a técnica SC parece ser mais eficiente e eficaz do que a inspeção AH. Isso é devido à orientação que a técnica *SCENARI_{IoT}CHECK* fornece aos inspetores sobre as características específicas da IoT na descrição dos cenários.

Este é um resultado que indica a adequação e aderência da técnica de inspeção aos princípios de técnicas de inspeções baseadas em *checklist*, que encapsula o conhecimento sobre uma perspectiva particular de um artefato e qualidade e possíveis raciocínios aplicáveis para lidar com as situações de design que podem levar a defeitos.

4.2.5 Ameaças a Validade e Limitação do Estudo

Todo estudo experimental possui sua limitação e uma série de ameaças a validade. Algumas limitações e ameaças identificadas estão descritas a seguir.

Como limitação deste estudo de viabilidade, pode-se considerar que esta pesquisa teve um interesse inicial para introduzir uma técnica de inspeção baseada em *checklist* específica para avaliar cenários em IoT no contexto de Sistemas de Software Contemporâneos.

A pequena amostra de participantes e seus conhecimentos neste estudo limitaram a conclusão dos resultados. Ambos os fatores podem influenciar diretamente nos resultados do estudo. A falta de um *template* para descrever cenários de forma mais organizada também foi uma ameaça desse trabalho.

Em relação aos participantes possuem um baixo conhecimento em descrever cenários baseados em IoT (usando a técnica *SCENARI_{IoT}*), uma alternativa foi dividi-los em grupos para que eles realizassem um *brainstorming* e tivessem mais ideias para resolver o problema proposto. Mesmo com treinamento, não é possível garantir que os participantes utilizaram adequadamente a técnica de *SCENARI_{IoT}*.

Juntamente com esse estudo de viabilidade, algumas ameaças à validade foram identificadas. Em relação à **validade de constructo**: não houve controle dos artefatos produzidos durante a execução do estudo, devido ao pouco tempo que o estudo foi executado. Não foi possível garantir que os cenários (artefatos) produzidos fossem comparáveis (equivalentes) quanto ao número de defeitos e complexidade de todos os cenários. No entanto, os grupos tinham conhecimentos semelhantes sobre IoT, obtidos por meio de um documento de caracterização de participantes.

A ameaça à **validade externa**: esse tipo de ameaça leva em consideração a generalização dos resultados. Alunos de graduação foram participantes neste estudo. Para (CARVER, JACCHERI, *et al.*, 2003), mesmo que os alunos de graduação não tenham experiência em aplicações industriais, eles ainda podem apresentar habilidades similares aos engenheiros de software iniciantes. Outra ameaça deste estudo, foi ele ter sido realizado em ambiente acadêmico, no entanto, o problema proposto pelos pesquisadores, foi uma situação real como a automação de uma estufa.

Sobre a **validade da conclusão**: o tamanho reduzido da amostra foi um fator ameaçador nesse estudo, o que limita a generalização e conclusão dos resultados

obtidos. Além disso, por uma questão de objetividade e correção, apenas a estatística descritiva foi utilizada para comparar os resultados. Para isso, outros estudos experimentais foram realizados para obter mais indicações sobre o desempenho da técnica *SCENARI_{IoT}CHECK*. A homogeneidade da amostra também é uma ameaça, devido ao tamanho da amostra. Portanto, não podemos dizer que os resultados sejam conclusivos. No entanto, podemos ter alguma indicação na aplicação da técnica *SCENARI_{IoT}CHECK* em cenários baseados em IoT.

Em relação à **validade interna**: nesse estudo, duas ameaças representaram um risco de interpretação inadequada dos resultados: treinamento e tempo de inspeção medido. Em relação ao treinamento, foi preparado o treinamento tanto para a inspeção *ad-hoc* quanto para a técnica *SCENARI_{IoT}CHECK*, bem como aplicamos um estudo piloto para observar a equivalência no treinamento e compreensão dos materiais utilizados. Foi solicitado que os grupos relatassem o tempo inicial e final de inspeção no documento de “registro de inspeção” (Apêndice C) para realizar os devidos cálculos.

4.3 Evolução da Técnica *SCENARI_{IoT}CHECK*

Com os resultados quantitativos e qualitativos, a técnica *SCENARI_{IoT}CHECK* foi evoluída para uma segunda versão, com a adição de novas perguntas e melhoria de outras. Após análise detalhada da técnica, verificou-se que: (1) as perguntas da *SCENARI_{IoT}CHECK* relacionadas as características em comum poderiam ser ordenadas e reagrupadas para se tornarem mais lógicas. (2) perguntas que avaliam as mesmas características (redundantes) poderiam ser excluídas. (3) algumas perguntas precisaram ser reformuladas, para tornarem-se mais compreensíveis e evitando assim ambiguidade nelas.

Em relação ao item (1), as perguntas relacionadas as mesmas características de avaliação foram redirecionadas e agrupadas para formar um conjunto de perguntas lógicas, visando reduzir o tempo de inspeção. No total, 32 perguntas foram agrupadas, onde onze conjuntos foram formados: cinco conjuntos em relação ao objetivo do sistema (coleta de dados, atores, semântica, cenários junto aos arranjos) e outros seis cenários em relação as facetas. A Figura 19 mostra um trecho da evolução da técnica *SCENARI_{IoT}CHECK* para uma nova versão. A Planilha (Apêndice L) mostra todas as categorias das perguntas organizadas por cores.

Em relação ao item (2), as perguntas redundantes foram excluídas, uma vez que não fazia sentido realizar uma mesma pergunta duas vezes para a mesma característica inspecionada, além de economizar tempo de inspeção. Quanto ao item (3), perguntas ambíguas foram reformuladas para se tornarem mais claras e objetivas, como “Os cenários buscam ser precisos?” para “Os cenários buscam ser precisos apresentando título e fluxos?”. A Figura 19 mostra exemplos de perguntas ambíguas e redundantes e suas respectivas evoluções.

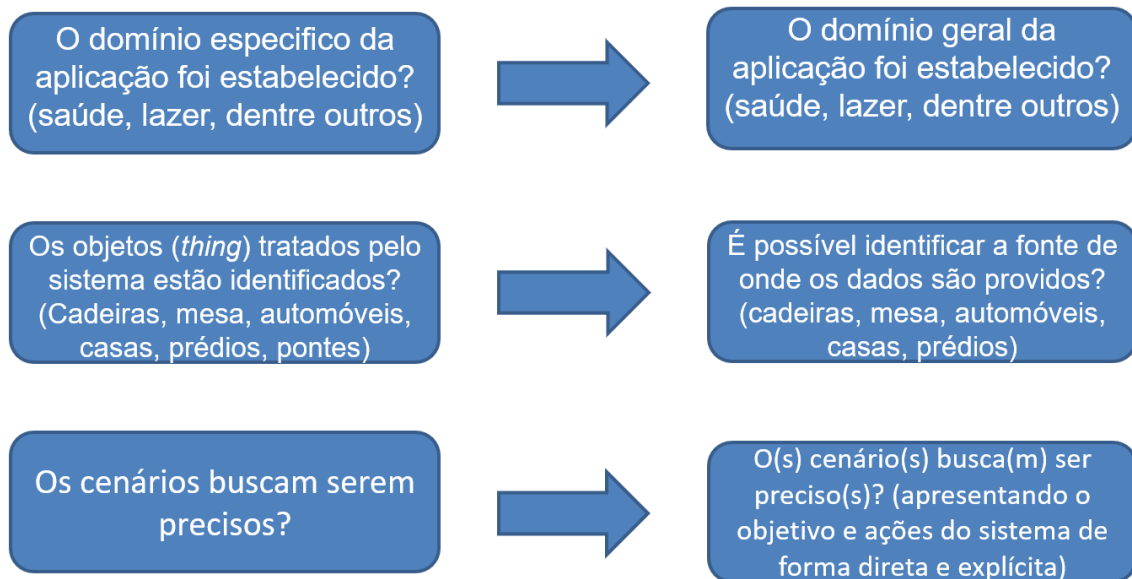


Figura 19. Trecho da evolução da Técnica SCENARI_{IoT}CHECK

4.4 Considerações Finais sobre o Capítulo

Os resultados quantitativos e qualitativos deste estudo fornecem subsídios essenciais para melhorar a técnica SCENARI_{IoT}CHECK. Em relação à análise quantitativa, demonstrou-se que os indicadores de eficácia e eficiência da técnica SC foram superiores aos indicadores de custo-eficiência, eficiência e eficácia referentes à técnica AH. No entanto, a comparação de ambas as técnicas não era o objetivo deste estudo. Pode-se notar que a técnica SC teve sucesso em alcançar seu objetivo, uma vez que ela conseguiu encontrar mais defeitos que a técnica AH.

Respondendo à pergunta principal: “A técnica SCENARI_{IoT}CHECK apoia a identificação de defeitos nas descrições de cenários para sistemas de software baseados em IoT?” Sim, o objetivo da abordagem foi alcançado. Os inspetores tiveram uma boa experiência e desempenho na aplicação da técnica, de acordo com os

resultados qualitativos e quantitativos. A técnica SC pode apoiar a composição do método para especificar artefatos de descrição de cenários de alta qualidade no contexto da engenharia de sistema de software IoT.

Além disso, a técnica *SCENARI_{IoT}CHECK* é a primeira técnica de leitura baseada em *checklist* e com indicação de viabilidade para apoiar a inspeção de artefatos de software em IoT, particularmente, as descrições de cenários baseadas em IoT produzidas com a técnica *SCENARI_{IoT}* (SOUZA, MOTTA, *et al.*, 2019).

5 Avaliando a Segunda Versão da SCENARI_{OT}CHECK

Neste capítulo é apresentado o estudo de observação conduzido com a segunda versão da técnica SCENARI_{OT}CHECK. O resultado desse estudo indica que a SCENARI_{OT}CHECK apresenta bom desempenho e atinge seus objetivos.

5.1 Introdução

Uma vez que os resultados obtidos com o estudo anterior (Capítulo 4) (SOUZA, MOTTA, *et al.*, 2019) indicaram a viabilidade da técnica SCENARI_{OT}CHECK, foi possível seguir com a segunda etapa da metodologia proposta por Shull, Carver e Travassos (2001), a execução de um estudo de observação. O propósito deste estudo é obter dados a partir da observação de como as tarefas da técnica são executadas pelos participantes. Geralmente, esse tipo de estudo ocorre em um ambiente controlado, porém não obrigatório, onde indivíduos desempenham tarefas enquanto são observados pelos pesquisadores.

A obtenção dos dados deste estudo se deu por meio de observação direta e coleta de dados a partir de artefatos, como: artefato de detecção de discrepâncias, caracterização dos participantes e questionário pós-inspeção. Neste estudo, os pesquisadores não interferiram enquanto os participantes executavam as tarefas, permitindo observar a segunda versão da técnica SCENARI_{OT}CHECK sob a perspectiva direta dos participantes.

5.2 Estudo de Observação

5.2.1 Planejamento

Nesta etapa, o design do estudo com todos os artefatos produzidos que foram utilizados pelos participantes e pesquisadores foi preparado (Apêndice F). Almejando observar como os inspetores se comportam utilizando a segunda versão da SCENARI_{OT}CHECK (Apêndice G), a questão de pesquisa proposta nesse estudo é “As etapas do processo de aplicação da SCENARI_{OT}CHECK fazem sentido?”

Para isso, o paradigma GQM (BASILI, CALDIERA, *et al.*, 1994) foi utilizado novamente para organizar a questão de pesquisa com objetivos e métricas. O estudo de observação tem o objetivo de **analisar** a técnica *SCENARI_{IoT}CHECK* **com o propósito de** compreender **em relação a** utilização da técnica (etapas, tarefas, e ordem de aplicação) **do ponto de vista** dos inspetores **no contexto de** desenvolvedores novatos realizando a inspeção de uma descrição de cenário de um sistema de software IoT.

O estudo foi realizado durante uma disciplina de engenharia de requisitos em uma turma de graduação no segundo semestre de 2019 na Universidade Federal do Amazonas (UFAM). Durante a disciplina, os tópicos discutidos foram: i) sistemas de software IoT: definições, aplicações e exemplos; e ii) especificações e inspeções de cenários para sistemas de software IoT, bem como suas aplicações. A taxonomia de defeitos (SHULL, Forrest Joseph, 1999) e processo de inspeção (KALINOWSKI, TRAVASSOS, 2004) utilizadas também foram a mesma que no estudo de viabilidade.

Inicialmente, 14 participantes indicaram que iriam participar do estudo. Porém, quatro participantes desistiram da disciplina e, conseqüentemente, do estudo, fazendo com que dez participantes estivessem disponíveis para sua execução. Todos os participantes tinham conhecimento em engenharia de software, uma vez que já haviam cursado duas disciplinas anteriormente, uma de engenharia de software e outra de desenvolvimento de software.

As variáveis dependentes consideradas nesse estudo foram as mesmas que no estudo de viabilidade (Capítulo 4): custo-eficiência, eficiência, eficácia, número de defeitos e tempo. Em relação às variáveis independentes, foram consideradas a experiência e o conhecimento dos participantes. O ambiente acadêmico foi selecionado neste segundo estudo, indicado adequado para observar como os inspetores aplicam a técnica (SHULL, Forrest, CARVER, 2001).

Instrumentação: as técnicas *SCENARI_{IoT}* e *SCENARI_{IoT}CHECK* foram utilizadas neste estudo. Foi elaborado um Termo de Consentimento, um documento de caracterização dos participantes, documento de registro de discrepância, o problema juntamente com os cenários, além do questionário pós inspeção com a técnica *SCENARI_{IoT}CHECK*. Além disso, um guia para descrição de cenários também foi proposto. Este guia foi derivado da técnica *SCENARI_{IoT}*. Cada participante recebeu um artefato contendo o domínio do problema, o cenário proposto para descrição, e treinamento sobre ambas as técnicas *SCENARI_{IoT}* e *SCENARI_{IoT}CHECK*. A princípio,

foi realizado um estudo piloto para observar a qualidade desses materiais, principalmente do guia (novo) proposto.

Piloto: O estudo piloto foi conduzido na disciplina de “Desenvolvimento de Software Orientado a Objetos” do curso de Engenharia de Informação e Computação da Poli/UFRJ, com seis alunos de graduação. Todos os participantes declararam ter experiência em desenvolvimento de software, uma vez que todos já haviam cursado outras disciplinas relacionadas ao desenvolvimento de software, incluindo Engenharia de Software. Na disciplina foi apresentado o problema para os estudantes, e depois, eles deveriam utilizar a técnica de especificação *SCENARIoT*. Para isso, os estudantes de graduação receberam o guia proposto para descrição dos cenários juntamente com a técnica *SCENARIoT*. Nesse estudo piloto, os participantes não realizaram a inspeção de software, devido a terem criado um único documento. Entretanto, a inspeção foi posteriormente realizada por dois estudantes de mestrado e mais um pesquisador de pós-doutorado do Grupo de Engenharia Experimental da COPPE/UFRJ. Após a inspeção, os pesquisadores responsáveis pelo estudo realizaram a correção e evolução nos instrumentos.

5.2.2 Execução

Os participantes assinaram o termo de consentimento e preencheram o formulário de caracterização. Então, foi solicitado que os participantes executassem as tarefas propostas. Com a obtenção da caracterização dos participantes, foi possível obter uma visão geral do conhecimento e experiência de cada participante. A Tabela 10 apresenta a informação sobre o conhecimento dos participante em relação a: descrição e inspeção de cenários, e sistemas IoT (Apêndice K).

Tabela 10. Caracterização dos participantes

Participante	Conhecimento em descrição de cenários	Conhecimento em inspeção	Conhecimento sobre sistemas de software IoT
1	Médio	Nenhum	Baixo
2	Nenhum	Nenhum	Nenhum
3	Médio	Baixo	Baixo
4	Baixo	Baixo	Baixo
5	Médio	Médio	Nenhum
6	Médio	Médio	Baixo
7	Baixo	Médio	Baixo
8	Médio	Médio	Baixo
9	Médio	Médio	Baixo
10	Médio	Médio	Baixo

O problema proposto neste estudo foi um sistema de software IoT para monitoramento de animais em uma fazenda. A ideia era construir um sistema que monitorasse a saúde do animal e sua distância da fazenda. Os cenários deveriam ser descritos usando a técnica *SCENARI_{IoT}* com apoio do guia proposto. Os participantes receberam um treinamento de como usar a técnica *SCENARI_{IoT}* e depois a aplicaram para descrever os cenários utilizando o guia. Cada participante gerou uma descrição de cenário, totalizando assim, dez diferentes cenários IoT para serem inspecionados. Os participantes descreveram os cenários em casa e os entregaram através do *Google Classroom*⁸.

Após a entrega dos cenários, os participantes receberam um treinamento sobre inspeção de software num contexto geral. E então, cada um recebeu um cenário (descrito por outro participante) para realizar a inspeção utilizando a técnica *ad-hoc*. Novamente, os participantes entregaram a tarefa pelo *Google Classroom*. O último treinamento apresentou a técnica *SCENARI_{IoT}CHECK*. Nesta etapa, os participantes receberam o mesmo cenário que o da inspeção usando a técnica *ad-hoc* para realizarem a inspeção usando a *SCENARI_{IoT}CHECK*. A intenção foi permitir aos participantes adquirir experiência no uso da técnica. Por fim, uma última rodada de inspeção foi realizada. Nessa, os participantes aplicaram novamente a técnica *SCENARI_{IoT}CHECK*, porém em cenários diferentes do que utilizaram nas duas primeiras rodadas de inspeção e ainda distinto do cenário que haviam inicialmente descrito. A Tabela 11 mostra como foi organizado o arranjo de distribuição (aleatório) do estudo.

Tabela 11. Distribuição dos participantes por inspeção

Participante	1º inspeção – AH	2º inspeção - SC	3º inspeção - SC
1	Cenário 7	Cenário 7	Cenário 10
2	Cenário 8	Cenário 8	Cenário 9
3	Cenário 9	Cenário 9	Cenário 8
4	Cenário 10	Cenário 10	Cenário 7
5	Cenário 6	Cenário 6	Cenário 1
6	Cenário 5	Cenário 5	Cenário 2
7	Cenário 4	Cenário 4	Cenário 3
8	Cenário 3	Cenário 3	Cenário 4
9	Cenário 2	Cenário 2	Cenário 5
10	Cenário 1	Cenário 1	Cenário 6

⁸ www.classroom.google.com

5.2.3 Resultados Quantitativos

Considerando o objetivo desse estudo observacional, não é possível fazer uma comparação direta entre as técnicas *ad-hoc* (AH) e *SCENAR_{IoT}CHECK* (SC). No entanto, é possível mostrar ambas as distribuições e fornecer linhas de base referentes à técnica SC. Os resultados obtidos neste estudo observacional foram analisados sob várias perspectivas, dentre elas: *tempo*, *discrepâncias*, *falsos positivos*, *defeitos*, *custo/eficiência* (CE), *eficiência* (EFF) e *eficácia* (EFC).

Todos os participantes identificaram um total de 108 defeitos no conjunto de cenários IoT utilizando ambas as técnicas AH e SC. Quatorze defeitos foram encontrados em comum (duplicatas), totalizando 94 defeitos conhecidos em todos os artefatos. Nossa intenção neste estudo foi observar como a segunda versão da técnica SC foi utilizada para executar a inspeção. O foco estava no desempenho e na evolução do participante em relação ao uso da técnica em si, não para comparar as técnicas AH e SC. O uso das métricas mencionadas acima (EFF, EFC e CE) é um substituto para compreender como os participantes aplicaram a técnica SC.

O resumo dos resultados deste estudo é apresentado na Tabela 12, onde podem ser observados os resultados das técnicas AH e SC relativos aos defeitos detectados, média de tempo, EFF, EFC e CE. Como pode ser observado, o comportamento foi semelhante ao observado no primeiro estudo, indicando que a evolução da técnica SC, a princípio, não afetou o comportamento anterior e manteve a viabilidade de sua utilização.

Tabela 12. Média dos indicadores obtidos

Rodada	Técnica	Defeitos Detectados	Tempo (H)	EFF	EFC	CE
1	AH	15	0,621	0,371	0,127	2,307
2	SC	45	0,643	0,642	0,396	7,954
3	SC	34	0,605	0,543	0,318	6,373

A Tabela 13 apresenta os resultados obtidos na primeira rodada de inspeção com a técnica AH quanto ao tempo, discrepâncias, falso positivo, defeitos detectados, eficiência, eficácia e custo-eficiência para cada participante.

Tabela 13. Primeira rodada de inspeção (utilizando técnica AH)

P	Tempo (H)	Discrepância	Falso Positivo	Defeitos Detectados	EFF	EFC	CE
1	0,233	2	2	0	0	0	0
2	1	7	5	2	0,286	0,222	2
3	0,666	3	3	0	0	0	0
4	0,133	2	2	0	0	0	0
5	0,983	7	4	3	0,429	0,2	3,052
6	0,716	4	3	1	0,25	0,167	1,397
7	1	3	1	2	0,667	0,2	2
8	0,633	4	1	3	0,75	0,231	4,737
9	0,516	3	1	2	0,667	0,154	3,876
10	0,3	3	1	2	0,667	0,100	6,006

A Tabela 14 apresenta os resultados obtidos da segunda rodada de inspeção utilizando a técnica SC quanto: ao tempo, discrepâncias, falso positivo, defeitos detectados, eficiência, eficácia e custo-eficiência para cada participante. Vale ressaltar que nessa segunda rodada de inspeção, os participantes inspecionaram o mesmo cenário que na primeira inspeção utilizando a técnica AH. Portanto, supõem-se que o aprendizado inicial tenha contribuído para aumentar o número de defeitos identificados. Entretanto, é possível observar que os inspetores puderam encontrar mais defeitos que o anteriormente indicado.

Tabela 14. Segunda rodada de inspeção (utilizando a técnica SC)

P	Tempo (H)	Discrepância	Falso Positivo	Defeitos Detectados	EFF	EFC	CE
1	1,066	3	2	1	0,333	0,5	0,938
2	0,833	8	1	7	0,875	0,778	8,403
3	0,316	4	0	4	1	0,4	12,658
4	0,716	7	4	3	0,429	0,158	4,2
5	0,566	14	4	10	0,714	0,667	17,668
6	0,183	4	3	1	0,25	0,167	5,464
7	1	5	2	3	0,6	0,3	3
8	0,483	4	1	3	0,75	0,231	6,211
9	0,666	7	3	4	0,571	0,154	6,006
10	0,6	10	1	9	0,9	0,1	15

A Tabela 15 apresenta os resultados obtidos da terceira rodada de inspeção utilizando a técnica SC quanto: ao tempo, discrepâncias, falso positivo, defeitos detectados, eficiência, eficácia e custo-eficiência para cada participante. Nessa terceira rodada de inspeção, os participantes inspecionaram cenários diferentes das rodadas anteriores, como mostra a distribuição da Tabela 11.

Tabela 15. Terceira rodada de inspeção (utilizando a técnica SC)

P	Tempo (H)	Discrepância	Falso Positivo	Defeitos Detectados	EFF	EFC	CE
1	0,683	2	1	1	0,5	0,25	1,464
2	0,85	7	2	5	0,714	0,556	5,882
3	0,733	2	1	1	0,5	0,1	1,364
4	0,133	5	4	1	0,2	0,5	7,519
5	0,416	14	5	9	0,643	0,450	21,63
6	0,566	5	1	4	0,8	0,308	7,067
7	0,666	7	3	4	0,571	0,4	6,006
8	0,633	2	1	1	0,5	0,167	1,580
9	0,4	4	2	2	0,5	0,133	5
10	0,966	12	6	6	0,5	0,316	6,211

A primeira perspectiva analisada nesse estudo foi eficiência. Considerando as três rodadas de inspeção, foi observada que a técnica teve uma aparente perda de eficiência quando comparada com o primeiro estudo. Porém, o mesmo comportamento foi observado com a técnica AH. O domínio do problema e o grande número de falsos positivos podem ter contribuído para este resultado, o que requer mais investigação para descobrir o motivo. A Figura 20 mostra o comportamento das técnicas AH e SC em relação aos dois estudos.

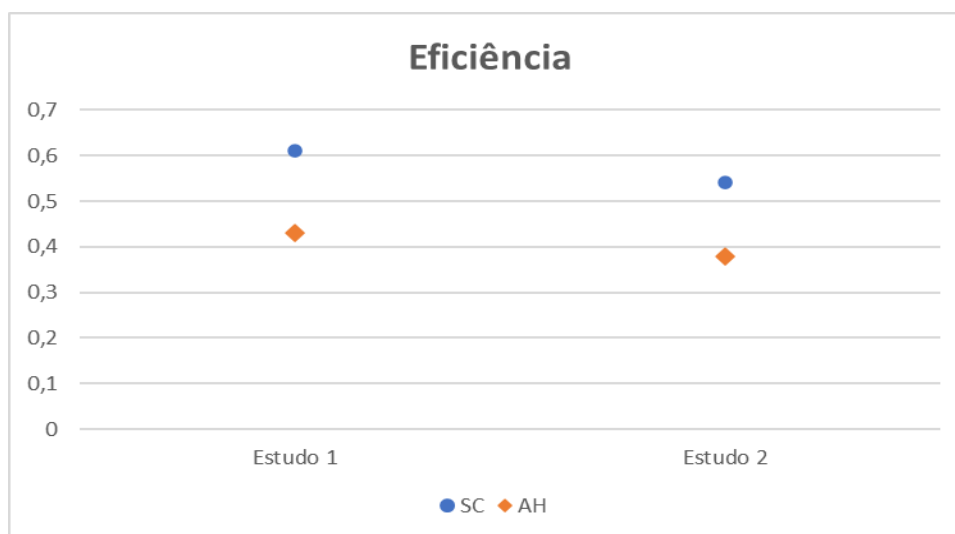


Figura 20. Comportamento das Técnicas AH e SC e ambos os estudos de viabilidade (1. estudo 1 – descrito no Capítulo 4) e observação (2. estudo 2 – rodada 1 descrito nesse capítulo) em relação a eficiência

A segunda perspectiva analisada foi eficácia; a técnica SC apresentou uma redução de desempenho mais intensa do que a técnica AH. Existem vários fatores de confusão, como cenários diferentes, pouca experiência e/ou nenhum conhecimento sobre a realização de uma inspeção dos participantes. Figura 21 mostra os diferentes

resultados na comparação das técnicas AH e SC na viabilidade e terceira rodada do estudo observacional.

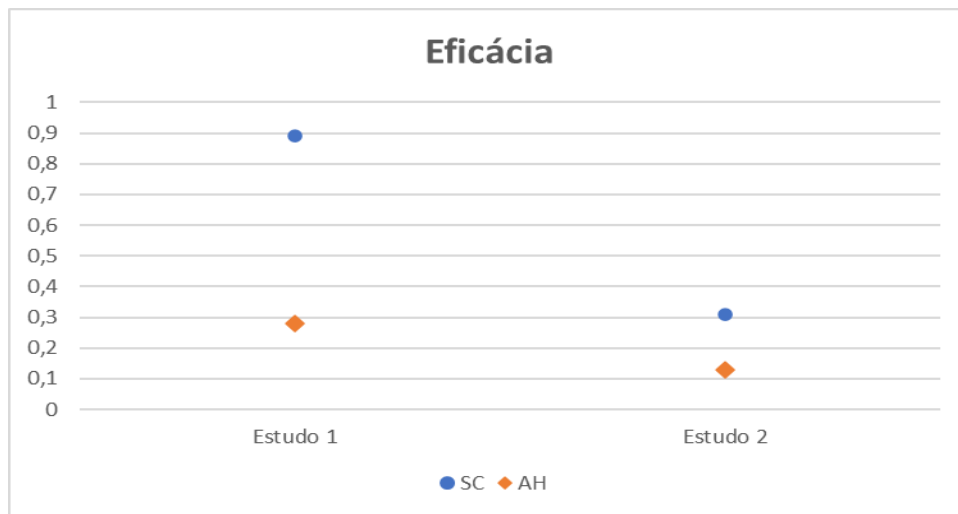


Figura 21. Comportamento das Técnicas AH e SC em ambos os estudos de viabilidade (estudo 1) e observação (estudo 2) em relação a eficácia

A terceira perspectiva analisada foi custo-eficiência; a qual apresentou melhores resultados que o primeiro estudo. No entanto, como podemos observar, mesmo a técnica SC apresentando uma redução de desempenho no segundo estudo, ela ainda supera a técnica AH em ambos os estudos. Figura 22 mostra os resultados em tempo, eficiência, custo-eficiência e eficácia.

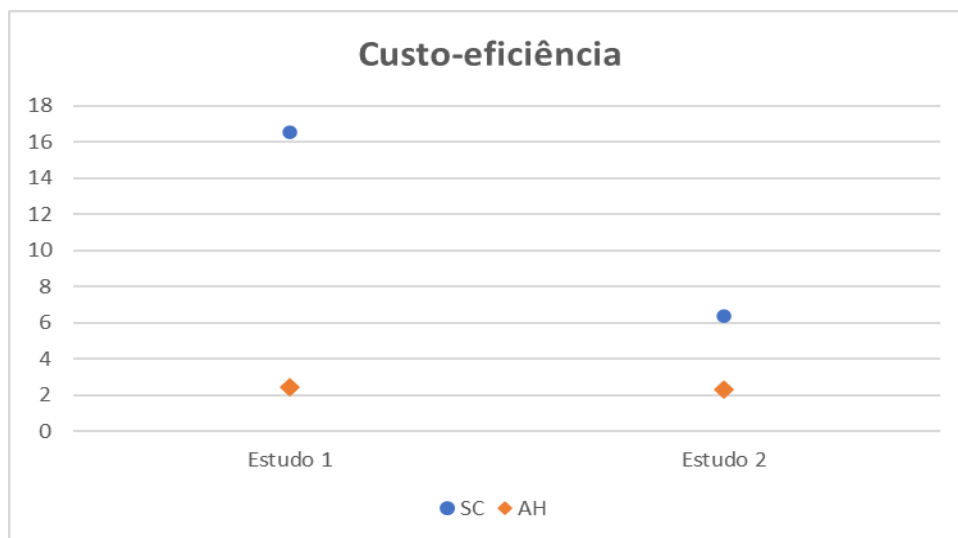


Figura 22. Comportamento das Técnicas AH e SC em ambos os estudos de viabilidade (estudo 1) e observação (estudo 2) em relação a custo-eficiência

A análise dos resultados baseou-se na evolução da técnica SC por meio de dois estudos experimentais e uma sequência de treinamentos e experiência dos participantes em inspeção de software. Nos dois estudos, pode-se observar que a técnica SC apresentou desempenho adequado quanto à eficiência, efetividade e custo-eficiência quando observado os resultados apresentados pela técnica AH. No entanto, a técnica SC apresenta um melhor desempenho no estudo de viabilidade quando comparado com o estudo observacional.

Este fato pode estar associado ao projeto e arranjo dos estudos. No estudo de viabilidade, os participantes foram divididos em grupos, enquanto no estudo observacional, os participantes realizaram a inspeção de maneira individual. Além disso, o domínio do problema e a representação de cenários foram apoiados por um modelo novo. Também pode-se citar a inserção de novas perguntas que a técnica adquiriu na segunda versão. Todas estas questões representam possíveis fatores de confusão que merecem ser observadas em estudos futuros. No entanto, mesmo incluindo todas essas alterações, a técnica SC realizou e apoiou a identificação de defeitos adequadamente.

Além disso, esses resultados também podem estar associados ao fato de a técnica SC orientar os inspetores para verificar as especificidades da IoT nos cenários, contribuindo para a sua aprendizagem. Portanto, deve-se levar em consideração o fator de aprendizagem (no estudo de observação), uma vez que os participantes realizaram uma inspeção utilizando a técnica AH e duas inspeções utilizando a técnica SC. Os participantes inspecionaram o mesmo cenário duas vezes (uma utilizando técnica AH e outra com a técnica SC). Por essa razão, não podemos fazer uma comparação entre as duas técnicas neste estudo observacional, pois a técnica AH estaria em desvantagem.

5.2.4 Resultados Qualitativos

Os resultados qualitativos serviram para verificar como os participantes aplicaram a *SCENARI_{IoT}CHECK* no contexto de uma inspeção de software. Após a aplicação da técnica SC, os participantes responderam a um questionário final sobre como a técnica influenciou no processo de inspeção de sistemas IoT.

O questionário aplicado aos participantes está disponível no Apêndice I. Assim como na análise do estudo de viabilidade (Capítulo 4), a análise dos resultados

qualitativos desse estudo se deu de forma *ad hoc*. Os dados sobre a *SCENARI_{IoT}CHECK* foram obtidos por meio de um questionário *pós-inspeção*.

Em relação ao grau de dificuldade da *SCENARI_{IoT}CHECK*, sete dos 10 participantes consideraram que a técnica é neutra em sua aplicação (Figura 23). Isso pode estar associado ao número de questões que a técnica possui. Além disso, podemos levar em consideração o treinamento dado aos participantes. O treinamento não garante que a técnica se torne mais fácil ou difícil.

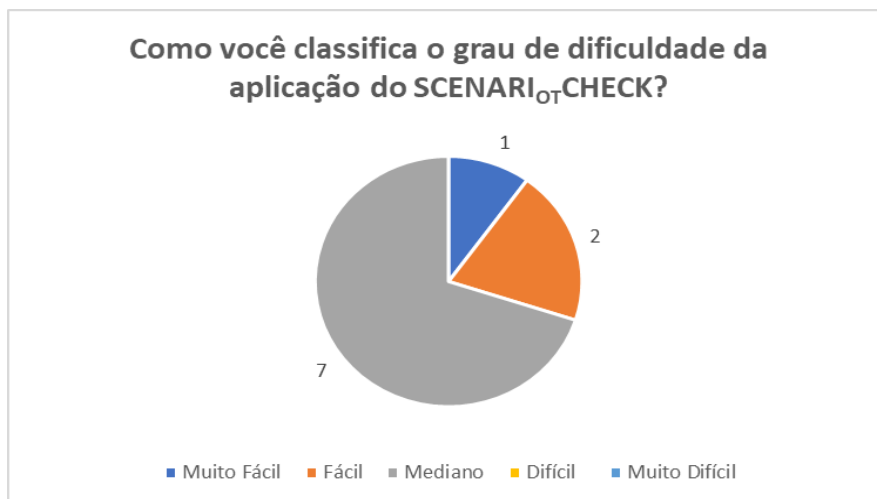


Figura 23. Percepção dos participantes em relação a dificuldade da aplicação da *SCENARI_{IoT}CHECK*

No contexto da facilidade/compreensão das perguntas da *SCENARI_{IoT}CHECK*, obteve-se resultados com algum desbalanceamento, onde quatro participantes relataram que a *SCENARI_{IoT}CHECK* possui perguntas fáceis enquanto outros quatro classificaram que a técnica tem perguntas medianas. Buscou-se sempre construir perguntas objetivas e de fácil compreensão para a técnica.

Além disso, um participante considerou as perguntas muito fáceis. E, por fim, outro considerou as perguntas difíceis. Esse resultado também pode estar associado ao grau de conhecimento dos participantes em relação as características novas de sistemas baseados em IoT. A Figura 24 mostra os resultados obtidos.

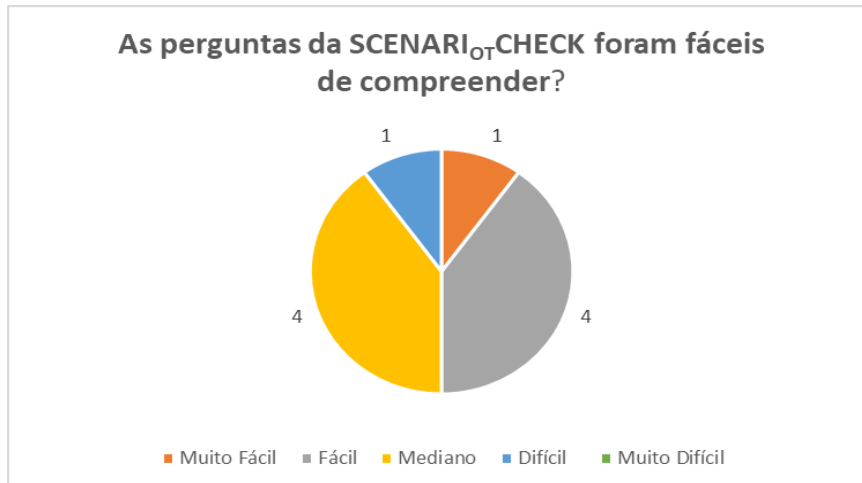


Figura 24. Percepção dos participantes em relação as perguntas da SCENARI_{OT}CHECK

Com relação a percepção do desempenho dos participantes em aplicar a SCENARI_{OT}CHECK. A Figura 25 apresenta os resultados alcançados com relação a percepção do desempenho dos participantes em aplicar a SCENARI_{OT}CHECK. Cinco dos participantes consideraram que a técnica ajudou a detectar defeitos nos cenários. Enquanto quatro concordam totalmente que a técnica tenha ajudado. Além disso, um participante relatou que a técnica não ajudou no momento da inspeção. Isso mostra como a técnica guia os inspetores a problemas específicos, uma vez que sistemas de software IoT utilizam outras características que sistemas *web* e *móvel*.

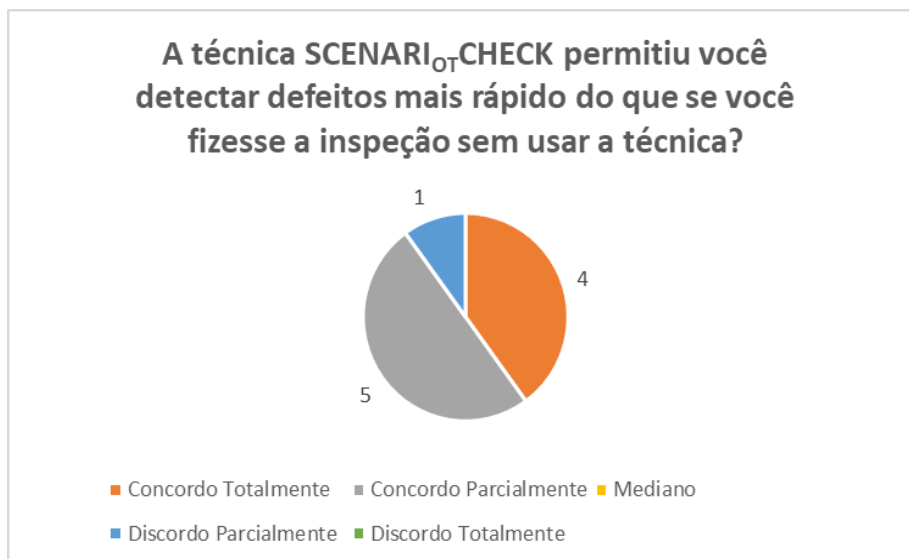


Figura 25. Percepção dos participantes em relação ao desempenho dele em aplicar SCENARI_{OT}CHECK

Além dos resultados dos questionários com perguntas fechadas, foi possível coletar algumas sugestões dos participantes com questões abertas. Em relação aos aspectos que tornam a técnica fácil/difícil: “Na sua opinião, quais aspectos da técnica tornam sua aplicação fácil/difícil de usar?” algumas das respostas foram:

“A forma como ocorre a interação da técnica com o cenário é muito boa, de forma simples e eficaz. Acredito que a parte mais difícil da técnica consiste em compreender as perguntas e tentar respondê-las pelo cenário, demandando mais tempo” Participante 4.

“A técnica em si é como uma fórmula que a pessoa pode seguir e encontrar a maior parte dos erros, pois de certa forma eles são padronizados e são cometidos de forma parecida pelas pessoas quando criam esses cenários” Participante 5.

“Os itens específicos e direcionados para análise no cenário” Participante 7.

“Fácil pois está bem especificado em grande parte. E difícil no sentido de se compreender o que está sendo pedido, mas nada tão difícil ao ponto de não entender”
Participante 9.

“Em relação a definição dos passos e em relação as perguntas, graças a elas tem se uma clareza maior sobre os defeitos nas especificações dos cenários, ajudando a encontrar os erros” Participante 10.

No contexto da técnica SC verificar todos os elementos e características de sistemas de software IoT, a seguinte pergunta buscou saber se a *SCENARI_{OT}CHECK* realmente inspecionava: “Você considera que as perguntas da técnica *SCENARI_{OT}CHECK* inspecionam todos os elementos dos cenários que são descritos? Se “sim” ou “não”, por quê?”

“Sim, por algumas perguntas serem bem amplas e específicas para todo o contexto dos cenários” Participante 2.

“Sim, abrangem todos os elementos descritos no cenário, pois seguiu uma sequência onde pude identificar cada parte do cenário avaliado através das perguntas presentes na técnica” Participante 4.

“Sim, pois a meu ver a técnica inspeciona os principais aspectos dos cenários apresentados e com isso o avaliador pode ver os elementos secundários ligados a estes e verificar todos os aspectos dos cenários” Participante 5.

“Sim, a técnica abrange todas as variáveis dos cenários descritos” Participante 6.

“Não, pois ainda podemos detalhar muito mais os itens a serem especificados”
Participante 7.

“Não, no geral a aplicação é boa, mas deveria ser mais especificada para cada título do cenário” Participante 8.

“Sim, pois estão bem escritas em boa parte da técnica” Participante 9.

“Sim, pois procura ver todas as interações entre os atores e atuadores, além de verificar questões referente a interação com o ambiente” Participante 10.

Por fim, foi feita uma pergunta a respeito de como as questões da técnica SC poderia ser melhorada: “Na sua opinião, como a técnica SCENARI_{OT}CHECK poderia ser melhorada?”

“Acredito que a técnica está no caminho certo, abrangendo todos os itens, do meu ponto de vista, que devem ser avaliados. Talvez uma melhor classificação dos defeitos ligados ao item específico do cenário, sendo mais fácil a identificação do dado a ser corrigido” Participante 4.

“Sendo ainda mais detalhista na hora de elaborar os questionários, pois pode haver pessoas que não entendam certos conceitos apresentados” Participante 5.

“Itens mais detalhados. Criar uma classificação para defeitos que não se encaixam em nenhuma Taxonomia” Participante 6.

“Ferramenta para identificar erro de contexto” Participante 7.

“A meu ver não notei defeitos, pois foi poucos dias usando a técnica”
Participante 8.

“Estava meio confusa para leigos, poderia ser bem mais detalhada. Poderia ter mais tipo na classificação dos erros” Participante 9.

A próxima Seção apresenta as ameaças identificadas nesse estudo de observação. Vale ressaltar que algumas ameaças identificadas são semelhantes ao estudo de viabilidade.

5.2.5 Ameaças a Validade

Quanto à **validade de constructo**, não houve controle na construção dos cenários durante o estudo de observação, devido ao pouco tempo que o estudo foi conduzido. Não foi possível garantir que os cenários produzidos fossem comparáveis em relação ao número de defeitos e complexidade. Pode-se apenas afirmar que os cenários eram relacionados ao sistema de software IoT. No entanto, os participantes tinham conhecimentos semelhantes sobre IoT e em como produzir os cenários IoT.

Em relação a **validade externa**: estudantes de graduação foram convidados a participar do estudo. Para Carver, Jaccheri *et al.*, (2003), embora alunos de graduação não tenham experiência em aplicações industriais, eles ainda podem apresentar habilidades semelhantes aos engenheiros de software iniciantes, mitigando essa ameaça. Outra ameaça foi realizar o estudo no meio acadêmico, porém esta ameaça foi mitigada levando em consideração uma situação real, no caso, a automação do monitoramento de animais em uma fazenda.

Sobre a **validade da conclusão**: Sobre a validade da conclusão: o tamanho da amostra limita naturalmente a generalização e conclusão dos resultados obtidos. Além disso, por uma questão de objetividade e correção, o erro tipo-I (alfa) foi ajustado para 0,10 juntamente com risco de erro tipo-II (beta), a partir da aplicação de testes não paramétricos. Nos próximos trabalhos, outros estudos experimentais serão realizados para obter mais indicações sobre o desempenho da técnica SC, principalmente estudos com profissionais da indústria.

Quanto à **validade interna**: o treinamento foi preparado tanto para a técnica *ad hoc* quanto para a *SCENARI_{IoT}CHECK*, assim como foi aplicado um estudo piloto para observar a equivalência nos treinamentos. Foi solicitado que os grupos relatassem o tempo inicial e final para ser possível a medição e ter uma média de quanto tempo foi necessário para executar uma inspeção. No entanto, ainda é um desafio garantir a consistência dessas medidas. Os resultados não podem ser generalizados e conclusivos, uma vez que a técnica ainda está sendo submetida a avaliações experimentais. É também importante ressaltar que este estudo foi realizado de forma assíncrona, sem o controle de variáveis de contexto, tendo em vista que os participantes realizaram as inspeções em suas casas.

5.3 Considerações Finais sobre o Capítulo

Esse Capítulo apresentou um estudo de observação conduzido com estudantes de graduação, cujo principal objetivo foi observar como a técnica SC é aplicada em uma inspeção de software. Para isso, o estudo foi projetado para que houvesse três sessões de inspeções, uma utilizando a técnica AH e duas utilizando a técnica SC. Observando os resultados de ambas as técnicas AH e SC, nota-se que a técnica SC detectou mais defeitos nos cenários IoT do que a técnica AH. Quando se compara os resultados da técnica SC em ambos os estudos de viabilidade (Capítulo 4) e observação, percebe-se que a primeira versão da técnica SC obteve resultados um pouco melhores do que sua segunda versão. Isso pode estar associado a evolução da técnica, dado que a técnica SC sofreu ajustes e evolução em sua estrutura como reorganização e inserção de novas perguntas.

Em relação à questão de pesquisa: “Os passos do processo fazem sentido?” Sim, os participantes aplicaram a técnica *SCENARIoT-CHECK* como esperado. O desempenho dos participantes foi melhor em relação a detecção de defeitos usando a técnica SC do que usando a técnica AH em ambos os estudos (observe que no segundo estudo, uma sessão de inspeção sugere que os inspetores detectaram mais defeitos com a técnica SC no mesmo artefato do que identificado com a técnica AH). Para apoiar os participantes na descrição dos cenários IoT, foi fornecido um modelo para sua descrição (complemento à *SCENARIoT*), que também está associado a técnica SC (DE SOUZA, MOTTA, *et al.*, 2019).

6 Considerações Finais

Este capítulo discute os principais resultados diante da condução dos estudos experimentais para avaliar a abordagem proposta, bem como as considerações finais deste estudo. Também são apresentadas as contribuições e perspectivas futuras.

6.1 Considerações Finais

As inspeções de software têm levado a benefícios consideráveis na área de engenharia de software, apoiando a leitura de artefatos produzidos durante as etapas de construção do software. Vários estudos experimentais realizados mostram os benefícios de técnicas de leitura para apoiar a identificação de defeitos, principalmente nas fases iniciais do projeto de software.

Diversas técnicas de inspeções têm sido propostas para avaliar artefatos de sistemas de software convencionais, como web e *mobile*. No entanto, quando mudamos de paradigma, por exemplo, para sistemas de software baseados em IoT, este cenário se transforma. Uma vez que não há técnicas de leitura para avaliar artefatos destes tipos de sistemas (Capítulo 2).

Essa dissertação apresentou uma nova técnica de inspeção, *SCENARI_{IoT}CHECK* (*Checklist for Reading IoT based Scenarios*), baseada em *checklist* para avaliar artefatos de sistemas de software IoT e que apoia a garantia da qualidade dos cenários produzidos pela *SCENARI_{IoT}*. Essa técnica foi criada para ser utilizada pelos engenheiros de software com baixo ou sem conhecimento em IoT em projetos de desenvolvimento de sistemas de software IoT.

A técnica tem o objetivo de ser direta e abrangente, para contribuir para a melhoria de artefatos baseados em IoT e, conseqüentemente, garantir a qualidade de sistemas de software IoT. Dois estudos experimentais foram realizados para observar sua viabilidade e utilização. Primeiro, foi realizado um estudo de viabilidade, a partir do qual os participantes afirmaram que a primeira versão da técnica não é difícil de utilizar e apoia a compreensão e a descoberta de defeitos. O resultado desse estudo apoiou a identificação de oportunidades de melhoria na técnica *SCENARI_{IoT}CHECK* (SOUZA, MOTTA, *et al.*, 2019).

Em seguida, foi realizado um estudo observacional com o objetivo de compreender como os inspetores aplicam a técnica SCENARI_{OT}CHECK. A ideia foi observar a adequação geral do processo de inspeção e dos artefatos propostos, considerando o desempenho individual dos participantes. A análise considerou os mesmos indicadores de eficiência, eficácia e custo-eficiência, tempo e defeitos presentes nos artefatos, considerando o conhecimento adquirido em cada rodada de inspeção (DE SOUZA, MOTTA, *et al.*, 2019).

Os estudos experimentais realizados nesse trabalho indicaram que a técnica SCENARI_{OT}CHECK obteve uma boa relação custo-eficiência, apresentou eficácia e eficiência satisfatórias quando aplicada de maneira adequada.

6.2 Contribuições

As contribuições gerais obtidas nesse estudo estão listadas de forma objetiva:

- Concepção e organização de uma técnica de inspeção (SCENARI_{OT}CHECK) no contexto de sistemas de software baseados em IoT, com avaliação experimental de sua viabilidade em apoiar a identificação de defeitos e garantir a qualidade dos cenários produzidos pela SCENARI_{OT}.
- Estabelecimento das características e propriedades de sistemas de software baseados em IoT que necessitam ser capturadas na especificação de requisitos do sistema de software.
- Criação de um guia (*template*) para apoiar a descrição de cenários com a SCENARI_{OT} (APÊNDICE M).

6.3 Limitações da Pesquisa

As principais limitações desta dissertação são:

- A generalização da aplicação da SCENARI_{OT}CHECK. Uma vez que não se sabe se a técnica é capaz de avaliar cenários produzidos por outras técnicas, como entrevista, *brainstorm* e casos de uso.
- A falta de apoio computacional para automatização da técnica SCENARI_{OT}CHECK. Esse apoio permitiria tornar o processo de

aplicação de ambas as técnicas (*SCENARI_{OT}* e *SCENARI_{OT}CHECK*) mais eficiente e em um contexto mais dinâmico que a indústria costuma possuir.

- Uma vez que esta dissertação realizou apenas dois estudos experimentais proposto na metodologia de Shull, Carver e Travassos (2001), seria oportuna a execução de, ao menos, mais dois estudos: Estudos de Caso em ciclos de vidas e em projetos na indústria, para aumentar a confiança na relação custo-eficiência da *SCENARI_{OT}CHECK*.

6.4 Perspectivas Futuras

As perspectivas futuras levadas em consideração no ciclo de vida da *SCENARI_{OT}CHECK*:

- Realização de um estudo secundário para caracterização de novas especificidades e propriedades de sistemas de software IoT;
- Realização de novos estudos experimentais para melhoria e evolução da técnica *SCENARI_{OT}CHECK*; no âmbito real de desenvolvimento (SHULL, CARVER e TRAVASSOS 2001).
- Propor uma infraestrutura de apoio computacional para ambas as técnicas (*SCENARI_{OT}* e *SCENARI_{OT}CHECK*) a fim de melhorar a sua aplicabilidade e usabilidade. E realizar estudos experimentais com o apoio computacional proposto.
- Investigar e realizar a aplicação da *SCENARI_{OT}CHECK* em cenários produzidos por outras técnicas de especificação de requisitos (além de cenários) para observar a aplicação da técnica.

Referências

AHMED, B.S., BURES, M., FRAJTAK, K., CERNY, T. "Aspects of Quality in Internet of Things (IoT) Solutions: A Systematic Mapping Study", **IEEE Access**, v. 7, p. 13758–13780, 2019. DOI: 10.1109/ACCESS.2019.2893493. .

ALSHAZLY, A.A., ELFATATRY, A. M., ABOUGABAL, M.S. "Detecting defects in software requirements specification", **Alexandria Engineering Journal**, v. 53, n. 3, p. 513–527, 2014. DOI: 10.1016/j.aej.2014.06.001.

ATZORI, L., IERA, A., MORABITO, G. "The Internet of Things: A survey", **Computer Networks**, v. 54, n. 15, p. 2787–2805, 2010. DOI: <https://doi.org/10.1016/j.comnet.2010.05.010>.

AURUM, A., PETERSSON, H., WOHLIN, C. "State-of-the-art: software inspections after 25 years", **Software Testing, Verification and Reliability**, v. 12, n. 3, p. 133–154, 2002a. DOI: 10.1002/stvr.243.

AURUM, A., PETERSSON, H., WOHLIN, C. "State-of-the-art: Software inspections after 25 years", **Software Testing Verification and Reliability**, v. 12, n. 3, p. 133–154, 2002b. DOI: 10.1002/stvr.243. .

AZIZ, M.W., SHEIKH, A.A., FELEMBAN, E.A. "Requirement Engineering Technique for Smart Spaces", in **16th International Conference on Internet of things and Cloud Computing**. Cambridge, United Kingdom, ACM Press, 2016. p. 1–7. DOI: 10.1145/2896387.2896439.

BARCELOS, R.F. "Uma abordagem para inspeção de documentos arquiteturais baseada em checklist", **Universidade Federal do Rio de Janeiro**, dissertação de mestrado, p. 183, 2006.

BASILI, V.R., CALDIERA, G., ROMBACH, H.D. "The goal question metric approach", **Encyclopedia of software engineering**, v. 2, p. 528– 532., 1994.

BASILI, V.R., GREEN, S., LAITENBERGER, O., LANUBILE, F., SHULL, F., SØRUMGÅRD, S., ZELKOWITZ, M.V. "The empirical investigation of perspective-based reading", **Empirical Software Engineering**, v. 1, n. 2, p. 133–164, 1996. DOI: 10.1007/BF00368702.

BERNÁRDEZ, B., GENERO, M., DURÁN, A., TORO, M. "A controlled experiment for evaluating a metric-based reading technique for requirements

inspection". in **10th International Symposium on Software Metrics**, 2004. p. 257–268. DOI: 10.1109/METRIC.2004.1357908.

BOEHM, B. "Verifying and Validating Software Requirements and Design Specifications", **IEEE Software**, v. 1, n. 01, p. 75–88, jan. 1984. DOI: 10.1109/MS.1984.233702. .

CARROLL, J.M. (Org.). **Scenario-Based Design: Envisioning Work and Technology in System Development**. USA, John Wiley & Sons, Inc., 1995.

CARVER, J., JACCHERI, L., MORASCA, S., SHULL, F. "Issues in using students in empirical studies in software engineering education", **proceedings. 5th International Workshop on Enterprise Networking and Computing in Healthcare Industry**, 2003. p. 239–249. DOI: 10.1109/METRIC.2003.1232471.

CHEN, S., XU, H., LIU, D., HU, B., WANG, H.. "A Vision of IoT: Applications, Challenges, and Opportunities With China Perspective", **IEEE Internet of Things Journal**, v. 1, n. 4, p. 349–359, 2014. DOI: 10.1109/JIOT.2014.2337336. .

CHENG, B., JEFFERY, R. "Comparing inspection strategies for software requirement specifications", **proceedings of 1996 Australian Software Engineering Conference**, 1996. p. 203–211. DOI: 10.1109/ASWEC.1996.534137.

CONTE, T., MASSOLLAR, J., MENDES, E., TRAVASSOS, G.H.. "Usability Evaluation Based on Web Design Perspectives", **First International Symposium on Empirical Software Engineering and Measurement**, 2007. p. 146–155. DOI: 10.1109/ESEM.2007.30.

CONTE, T., TRAVASSOS, G. H. "Técnica de inspeção de usabilidade baseada em perspectivas de projeto Web", **Rio de Janeiro: UFRJ/COPPE**, 2009. .

DAR, H., LALI, M.I., ASHRAF, H., *et al.* "A systematic study on software requirements elicitation techniques and its challenges in mobile application development", **IEEE Access**, cited By 3, v. 6, p. 63859–63867, 2018. DOI: 10.1109/ACCESS.2018.2874981.

DE BRITO, J., TROVAO, J., DIAS-NETO, A. "TestCheck: uma abordagem baseada em Checklist para inspecionar artefatos de teste de software", in **Simpósio Brasileiro de Qualidade de Software**, 2012, p. 113-127.

DE SOUZA, B.P., MOTTA, R. C., DE O. COSTA, D., TRAVASSOS, G.H. "An IoT-Based Scenario Description Inspection Technique", **Proceedings of the XVIII**

Brazilian Symposium on Software Quality, New York, NY, USA, Association for Computing Machinery, 2019. p. 20–29. DOI: 10.1145/3364641.3364644.

LIAO, Y., DESCHAMPS, F., LOURES, E., RAMOS, L.. "Past , present and future of Industry 4.0 - a systematic literature review and research agenda proposal", **International Journal of Production Research**, n. January 2019, p. 0, 2017. DOI: 10.1080/00207543.2017.1308576.

DOS SANTOS, P.S.M., TRAVASSOS, G.H. "Inspeção de Qualidade em Descrições de Casos de Uso: uma Avaliação Experimental em um Projeto Real", in **IX Simpósio Brasileiro de Qualidade de Software**, 2010, p. 261-278.

EBAD, S.A. "Inspection reading techniques applied to software artifacts - A systematic review", **Computer Systems Science and Engineering**, v. 32, n. 3, p. 213–226, 2017. .

FAGAN, M.E. "Advances in Software Inspections", **IEEE Transactions on Software Engineering**, v. SE-12, n. 7, p. 744–751, 1986. DOI: 10.1109/TSE.1986.6312976. .

GAUR, A., SCOTNEY, B., PARR, G., McCLEAN, S. "Smart City Architecture and its Applications Based on IoT", **Procedia Computer Science, The 6th International Conference on Ambient Systems, Networks and Technologies (ANT-2015)**, the 5th International Conference on Sustainable Energy Information Technology (SEIT-2015), v. 52, p. 1089–1094, 2015. DOI: <https://doi.org/10.1016/j.procs.2015.05.122>.

GIUSTO, D., IERA, A., ATZORI, L. **The internet of things: 20th Tyrrhenian workshop on digital communications**. 20th. ed. Springer.

GLASS, R.L. **Software runaways-Lessons learned from massive software project failures. 1998**. Prentice Hall.

GÓMEZ-CHABLA, R., REAL-AVILÉS, K., MORÁN, C., GRIJALVA, P., RECALDE, T. "IoT Applications in Agriculture: A Systematic Literature Review", **Advances in Intelligent Systems and Computing**, cited By 0, v. 901, p. 68–76, 2019. DOI: 10.1007/978-3-030-10728-4_8.

HERNANDES, E.M., BELGAMO, A., FABBRI, S. "An Overview of Experimental Studies on Software Inspection Process", in **International Conference on Enterprise Information Systems**, Springer International Publishing, 2014. p. 118–134.

KALINOWSKI, M., TRAVASSOS, G.H. "A computational framework for supporting software inspections", **19th International Conference on Automated Software Engineering**, 2004. p. 46–55. DOI: 10.1109/ASE.2004.1342723.

LAITENBERGER, O., "A SURVEY OF SOFTWARE INSPECTION TECHNOLOGIES". **Handbook of Software Engineering and Knowledge Engineering**. p. 517–555. DOI: 10.1142/9789812389701_0023.

LANUBILE, F., SHULL, F., BASILI, V. R. "Experimenting with Error Abstraction in Requirements Documents". **proceedings Fifth International Software Metrics Symposium. Metrics**, 1998, p. 114-121. DOI: 10.1109/METRIC.1998.731236.

LEITE, J.C.S. do P., DOORN, J.H., HADAD, G.D.S., KAPLAN, G. "Scenario inspections", **Requirements Engineering**, v. 10, n. 1, p. 1–21, 2005. DOI: 10.1007/s00766-003-0186-9.

LIM, T.-Y., CHUA, F.-F., TAJUDDIN, B. B. "Elicitation Techniques for Internet of Things Applications Requirements", **proceedings of the 2018 VII International Conference on Network, Communication and Computing**, Taipei City, Taiwan, ACM Press, 2019. p. 182–188. DOI: 10.1145/3301326.3301360.

LIU, L., YU, E. "Designing information systems in social context: a goal and scenario modelling approach", **Information Systems**, The 14th International Conference on Advanced Information Systems Engineering (CAiSE*02), v. 29, n. 2, p. 187–203, 2004. DOI: [https://doi.org/10.1016/S0306-4379\(03\)00052-8](https://doi.org/10.1016/S0306-4379(03)00052-8).

MAHALANK, S. N., MALAGUND, K. B., BANAKAR, R. M. "Non functional requirement analysis in IoT based smart traffic management system", **International Conference on Computing Communication Control and automation**, 2017. p. 1–6. DOI: 10.1109/ICCUBEA.2016.7860147.

MAIDEN, N.A.M., "CREWS-SAVRE: Scenarios for Acquiring and Validating Requirements". In: SUTCLIFFE, A., BENYON, D. (Org.), **Automated Software Engineering**, Boston, MA, Springer US, 1998. v. 5. p. 419–446. DOI: 10.1023/A:1008605412971.

MARTIN, J., TSAI, W. T. "N-Fold inspection: a requirements analysis technique", **Communications of the ACM**, v. 33, n. 2, p. 225–232, 1990. DOI: 10.1145/75577.75587.

MOTTA, R., SILVA, V., TRAVASSOS, G. "Towards a more in-depth

understanding of the IoT Paradigm and its challenges", **Journal of Software Engineering Research and Development**, v. 7, p. 3:1--3:16, 2019. DOI: 10.5753/jserd.2019.14.

MOTTA, R.C., DE OLIVEIRA, K.M., TRAVASSOS, G. H. "On challenges in engineering IoT software systems", **proceedings of the XXXII Brazilian Symposium on Software Engineering**, New York, NY, USA, ACM, 2018. p. 42–51. DOI: 10.1145/3266237.3266263.

MOTTA, R.C., DE OLIVEIRA, K. M., TRAVASSOS, G. H. "A Framework to Support the Engineering of Internet of Things Software Systems", **proceedings of the ACM SIGCHI Symposium on Engineering Interactive Computing Systems**, New York, NY, USA, Association for Computing Machinery, 2019. DOI: 10.1145/3319499.3328239..

NARDI, B. A. "The use of scenarios in design", **ACM SIGCHI Bulletin**, v. 24, n. 4, p. 13–14, 2007. DOI: 10.1145/142167.142171.

PACHECO, C., GARCIA, I., REYES, M. "Requirements elicitation Techniques: A systematic literature review based on the maturity of the techniques", **IET Software**, cited By 7, v. 12, n. 4, p. 365–378, 2018. DOI: 10.1049/iet-sen.2017.0144.sen.2017.0144&partnerID=40&md5=1dc94bcf27e1f66d42c2fff649db581a.

PORTER, A.A., VOTTA, L.G. "An experiment to assess different defect detection methods for software requirements inspections", **proceedings of 16th International Conference on Software Engineering**, 1994. p. 103–112. DOI: 10.1109/ICSE.1994.296770.

REGGIO, G. "A UML-based proposal for IoT system requirements specification", **International Workshop on Modelling in Software Engineering**, Gothenburg, Sweden, ACM Press, 2018. p. 9–16. DOI: 10.1145/3193954.3193956.

SAHA, H.N., AUDDY, S., PAL, S., KUMAR, S., PANDEY, S., SINGH, R., SINGH, A., BANERJEE, S., GHOSH, D., SAHA, S. "Waste management using Internet of Things (IoT)", **8th Annual Industrial Automation and Electromechanical Engineering Conference**, 2017. p. 359–363. DOI: 10.1109/IEMECON.2017.8079623.

SAUER, C., JEFFERY, D., LAND, L., YETTON, P. "The effectiveness of software development technical reviews: A behaviorally motivated program of research", **IEEE Transactions on Software Engineering**, v. 26, n. 1, p. 1–14, 2000. DOI: 10.1109/32.825763. .

SHI, X., AN, X., ZHAO, Q., LIU, H., XIA, L., GUO, Y. "State-of-the-art internet of things in protected agriculture", **Sensors (Switzerland)**, cited By 11, v. 19, n. 8, 2019. DOI: 10.3390/s19081833.

SHULL, F., CARVER, J., TRAVASSOS, G.H. "An Empirical Methodology for Introducing Software Processes", **Proceedings of the 8th European software engineering conference held jointly with 9th ACM SIGSOFT international symposium on Foundations of software engineering**, p. 9, 2001.

SHULL, F.J. "Developing Techniques for Using Software Documents: A Series of Empirical Studies", **Thesis presented to University of Maryland**, 1999, p. 1-86.

SILVA, V.M. "SCENARIOT Support for Scenario Specification of Internet of Things-based Software Systems", **Universidade Federal do Rio de Janeiro**, dissertação de mestrado, 2019.

SOUZA, B.P., MOTTA, R.C., TRAVASSOS, G.H. "The first version of SCENARIOT CHECK A Checklist for IoT based Scenarios", **Proceedings of the XXXIII Brazilian Symposium on Software Engineering**, 2019.

THELIN, T., RUNESON, P., WOHLIN, C., OLSSON, T., ANDERSSON, C. "How much information is needed for usage-based reading? A series of experiments", **Proceedings International Symposium on Empirical Software Engineering**, 2002. p. 127–138. DOI: 10.1109/ISESE.2002.1166932.

TRAVASSOS, G., SHULL, F., FREDERICKS, M., BASILI, V. "Detecting defects in object-oriented designs: using reading techniques to increase software quality", **ACM Sigplan Notices**, v. 34, n. 10, p. 47–56, 1999.

WALIA, G.S., CARVER, J.C. "A systematic literature review to identify and classify software requirement errors", **Information and Software Technology**, v. 51, n. 7, p. 1087–1109, 2009. DOI: 10.1016/j.infsof.2009.01.004.

ZACHMAN, J.A. "A framework for information systems architecture", **IBM Systems Journal**, v. 26, n. 3, p. 276–292, 1987. DOI: 10.1147/sj.263.0276. .

ZAMBONELLI, F. "Key Abstractions for IoT-Oriented Software Engineering", **IEEE Software**, v. 34, n. 1, p. 38–45, 2017. DOI: 10.1109/MS.2017.3.

Apêndice A – Protocolo do Estudo de Viabilidade

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
PROGRAMA DE ENGENHARIA DE SISTEMAS DA COMPUTAÇÃO
ENGENHARIA DE SOFTWARE EXPERIMENTAL

TEMPLATE PARA PLANO DE ESTUDOS EXPERIMENTAIS

1. IDENTIFICAÇÃO

Título: SCENARI_{IoT}CHECK

Área Técnica: Engenharia de Software

Autores: Bruno Pedraça de Souza e Guilherme Horta Travassos

Afiliação: COPPE/UFRJ

Local: Rio de Janeiro

Data: 15 a 18 de janeiro

2. CARACTERIZAÇÃO

Será realizado um estudo de viabilidade aplicado em um minicurso sobre engenharia de requisitos.

Tipo: Estudo de viabilidade.

Domínio: Engenharia de Software Experimental – Especificação e Inspeção de Requisitos.

Língua: Português (Brasileiro).

Parceiros: Universidade Federal do Rio de Janeiro – PESC/COPPE e Universidade Federal do Amazonas - ICET/UFAM.

Expectativa de Execução: Primeiro Semestre de 2019

Glossário de Termos:

Engenharia de software (ES);

Internet das coisas (IoT);

3. INTRODUÇÃO

A análise e especificação de requisitos são etapas essenciais dentro da engenharia de software e são consideradas por muitos como uma das etapas mais críticas na construção de um produto, uma vez que especificar requisitos de maneira inadequada pode acarretar danos catastróficos para empresas desenvolvedoras de software. Juntando a isso, com a nova categoria de sistemas de software que estão surgindo, os sistemas IoT. Sistemas IoT são todos os novos tipos de sistemas, que de alguma forma são sistemas parciais ou totalmente autônomos. Esses novos sistemas se diferenciam

dos antigos, pelo fato da interação, maneira de construí-los e diferentes tipos de hardware, além da incorporação das “coisas” nos sistemas. A proposta inicial do estudo é observar a viabilidade se a técnica SCENARI_{OT}CHECK consegue capturar defeitos nos cenários especificados pela técnica de Martins (2019).

4. DEFINIÇÃO DO ESTUDO EXPERIMENTAL

Objeto de Estudo: SCENARI_{OT}CHECK.

Objetivo Global: Este trabalho tem o propósito de realizar um estudo de viabilidade com estudantes de graduação em engenharia de software para verificar se a técnica SCENARI_{OT}CHECK é capaz de detectar defeitos nos cenários descritos pela SCENARI_{OT}.

Objetivos Específicos

- **Analisar:** SCENARI_{OT}CHECK
- **Com o propósito de:** caracterizar
- **Em relação:** a viabilidade da SCENARI_{OT}CHECK, observada em termos de eficácia, custo-eficiência e eficiência.
- **Do ponto de vista:** dos pesquisadores em ES
- **No Contexto:** Uma inspeção do documento de cenários baseado em sistemas de software IoT por estudantes de graduação.

Questões e Métricas

O uso da técnica de inspeção SCENARI_{OT}CHECK é mais eficaz em encontrar defeitos que da técnica ad-hoc?

Esta questão tem como objetivo verificar, no contexto do *projeto IoT*, se a eficácia utilizando a técnica SCENARI_{OT}CHECK é melhor do que a técnica ad-hoc.

O uso da técnica de inspeção SCENARI_{OT}CHECK é mais eficiente em encontrar defeitos que da técnica ad-hoc?

Esta questão tem como objetivo verificar, no contexto do *projeto IoT*, se a eficiência utilizando a técnica SCENARI_{OT}CHECK é melhor do que a técnica ad-hoc.

O uso da técnica de inspeção SCENARI_{OT}CHECK possui maior custo-eficiência em encontrar defeitos que da técnica ad-hoc?

Esta questão tem como objetivo verificar, no contexto do *projeto IoT*, se o custo-eficiência utilizando a técnica SCENARI_{OT}CHECK é melhor do que a técnica ad-hoc.

- Custo-eficiência: razão entre o nº total de defeitos encontrados pelo tempo (h);
- Eficácia: razão entre defeitos detectados por cada grupo dividido pelo número total de defeitos conhecidos no artefato inspecionado;
- Eficiência: razão entre defeitos detectados por cada grupo dividido pelo número de discrepâncias
- Questionário: coleta da percepção de cada inspetor;

5. PLANEJAMENTO

Seleção de Variáveis

Variáveis dependentes: custo-eficiência, eficiência, eficácia e número de defeitos.

Variáveis independentes: tempo e a experiência dos participantes.

Não definimos uma hipótese, uma vez que este estudo teve uma pequena amostra. Portanto, a aplicação de testes estatísticos não seria adequada.

Seleção dos Participantes

Critério de Seleção de Participantes: amostra por conveniência

Experiência Necessária: os estudantes tenham cursado a disciplina de Engenharia de Software.

Critério de Seleção de Grupos: serão quatro grupos (todos os grupos utilizarão ambas as técnicas de inspeções: Ad hoc e SCENARI_{IoT}CHECK).

Técnicas Probabilísticas de Amostragem: não se aplica;

Técnicas não Probabilísticas de Amostragem: não se aplica.

Recursos:

- **Software:** artefato de descrição de cenários; técnica SCENARI_{IoT}CHECK
- **Hardware:** Um notebook para a apresentação do material e treinamento em sala de aula;
- **Questionários:** um questionário para coleta de dados qualitativos (após terem utilizado a técnica SCENARI_{IoT}CHECK), termo de concordância em participação do estudo e caracterização dos participantes.

Design do Experimento

- **Objetos:** artefato de requisitos baseados em cenários;
- **Medições:** eficiência, eficácia e percepção do inspetor;
- **Técnicas:** Ad hoc e SCENARI_{IoT}CHECK;

Instrumentação

Descrição da Instrumentação: nesta parte da pesquisa, as perguntas foram preparadas para os participantes quando eles forem responder o questionário. Será utilizada duas formas para obtenção/coleta dos dados: questionário e observação direta.

- **Apoio à Análise Quantitativa:** não será necessário;
- **Apoio à Análise Qualitativa:** análise ad hoc;
- **Critérios de Observação:** sem intervenção dos pesquisadores;
- **Artefatos (Questionários, Procedimentos, etc):** questionário;

Mecanismos de Análise

- **Critérios para Eliminação de *Outliers*:** não se aplica;

6. TREINAMENTO

Definição do Treinamento e Procedimentos

- **Aplicadores:** pelo pesquisador;
- **Participantes:** estudantes de graduação;

Procedimentos: aulas sobre IoT; engenharia de requisito; treinamento sobre descrição de cenários para sistemas de software IoT; treinamento sobre técnicas de

inspeções; treinamento sobre técnicas de inspeções para sistemas de software contemporâneos.

7. PROCEDIMENTOS DE EXECUÇÃO

Definição de Execução do Estudo Experimental: será realizado um estudo de viabilidade com estudantes de graduação dos cursos de engenharia de software e sistemas de informação. Será aplicado em um minicurso nas férias. Inicialmente, será realizado um minicurso sobre IoT e engenharia de requisitos (especificação e inspeção). Mostrando para os alunos quais as perspectivas de IoT que os pesquisadores estão considerando. Além disso, exemplificação de aplicações IoT. A parte de engenharia de requisitos será ministrada as aulas sobre especificação de requisitos, especificamente, técnicas de especificação como entrevista, cenários, e entre outros. Além disso, será realizado um treinamento sobre inspeção de requisitos de software, juntamente com aplicação prática. Após a aula, os alunos serão convidados a realizarem a atividade de inspeção nos documentos que serão dados a eles. No final, os alunos também serão convidados a responderem um questionário sobre a utilização da técnica SCENARI_{IoT}CHECK. O minicurso será realizado nos dias 15 a 18 de janeiro. Duas horas por aula.

- **Artefatos (Instruções, Documentos, etc):**
- **Termo de Consentimento Livre e Esclarecido:** para o respaldo e anonimato dos alunos.
- **Caracterização dos participantes:** para capturar a experiência dos alunos.
- **Questionário:** para coletar opinião dos participantes após a utilização da SCENARI_{IoT}CHECK.

8. AVALIAÇÃO DO PLANO

- **Objetivos:** revisado pelos autores;
- **Participantes:** pesquisadores;
- **Procedimentos de Execução:** 4 dias (seguindo este plano de execução);
- **Artefatos Utilizados:** técnicas SCENARI_{IoT} e SCENARI_{IoT}CHECK;
- **Artefatos Gerados (Lições Aprendidas, Sugestões de Modificação do Plano):** descrição de cenários para sistemas IoT (SCENARI_{IoT}); documento de registro de defeitos (para Ad hoc e SCENARI_{IoT}CHECK); documento de inspeção dos cenários com a SCENARI_{IoT}CHECK. E questionário de percepção dos participantes após o uso da técnica;

9. PLANEJAMENTO DE CUSTOS

Custos do Estudo Experimental

- **Custos de Planejamento:** não se aplica;
- **Plano em si:** não se aplica;
- **Instrumentação:** não se aplica;
- **Material de Treinamento:** não se aplica;

- **Avaliação do Plano:** não se aplica;
- **Custos de Execução**
- **Deslocamentos:** não se aplica;
- **Treinamentos:** será aplicado treinamento das técnicas (de descrição de cenários e inspeção para sistemas de software IoT) para os estudantes;
- **Recursos Humanos:** estudantes de graduação e pesquisadores;
- **Recursos Materiais:** formulários, computadores, software.
- **Custos de Análise:** não se aplica;
- **Custos de Empacotamento:** não se aplica;

Apêndice B - Primeira Versão da SCENARI_{OT}CHECK

#	Item de Inspeção	Questão	Sim	Não
1	Descrição Geral do Sistema	O objetivo do sistema está descrito detalhadamente? (Somente visualização de dados; visualização, tomada de decisão e atuação)		
2		O domínio específico da aplicação foi estabelecido? (Saúde, lazer, trânsito)		
3		O tipo de dado coletado está especificado? (Temperatura, umidade, poluição)		
4		É possível identificar quem ou o quê acessa os dados coletados? (Usuários, sistemas de software)		
5		Os papéis envolvidos com o sistema estão descritos? (Sistemas de software, usuários)		
6		É possível identificar quem ou o quê visualiza os dados? (Usuários, sistemas de software)		
7		Os atores descritos nos cenários estão de acordo com os atores descritos nos arranjos? (<i>Things</i> , usuários, sistemas de software)		
8		Existe alguma descrição detalhada de cada ator nos cenários especificados?		
9		É possível identificar “coisas” descritas com determinada função nos arranjos que representa outra função nos cenários?		
10		São evitados advérbios que gerem mais de uma possibilidade de interpretação nos cenários? (<i>Provavelmente, possivelmente, supostamente</i>)		
11		Quando se é utilizado uma palavra “coisas”, “dados” nos cenários, ela é utilizada com o mesmo significado em outros artefatos?		
12		Os cenários buscam serem precisos? (Procurando responder de forma direta o cenário)		
13		Os cenários buscam estarem relacionados aos arranjos?		

#	Item de Inspeção	Questão	Sim	Não
14	Ambiente	É possível identificar o contexto no qual o sistema está inserido? (Cidade inteligente, agricultura, automobilística, assistência em saúde)		
15		As limitações do ambiente estão descritas? (Exemplo: falta de estrutura de conectividade, falta de estrutura de hardware, infraestrutura inadequada)		
16	Coisas	É possível identificar quem ou o quê coleta os dados? (Sensores, leitores de QR code, pessoas)		
17		Os objetos (<i>thing</i>) tratados pelo sistema estão identificados? (Cadeiras, mesa, automóveis, casas, prédios, pontes)		
18		As tecnologias associadas aos objetos do sistema estão descritas? (<i>Smart phones, smart watches, wearables</i>)		
19	Comportamento	É possível identificar quem ou o quê gerencia os dados/informações coletadas?		
20		É possível identificar quem ou o quê realiza a ação ou atua no sistema? (Atuador, usuários)		
21		Os eventos que o sistema possui foram identificados? (Exemplo: ligar/desligar um objeto, enviar dados)		
22	Conectividade	O tipo de tecnologia de comunicação que o sistema utiliza está descrito nos cenários? (<i>Bluetooth, intranet, internet...</i>)		
23		A tecnologia de comunicação proposta atende as especificações geográficas/físicas do sistema? (Larga, média ou pequena escala)		
24	Inteligência	É possível identificar como o sistema reage de acordo com as alterações ocorridas no ambiente?		
25	Interatividade	As interações do sistema com o ambiente estão representadas nos cenários?		

Apêndice C - Registro de Inspeção – Versão 1

REGISTRO DE INSPEÇÃO						
Data da inspeção: Responsável pela inspeção: Hora inicial: Hora final: Tempo despendido: (hh:mm)						
Registros de não-conformidades						
	N/A	Sim	Não	Cenário	Descrição (Localização, obs.)	Tipo de defeito
Observações:						

Apêndice D - Caracterização das questões da SCENARI_{OT}CHECK

Item de Avaliação - Questão 1	
Descrição do Item	O domínio geral da aplicação foi estabelecido? (saúde, lazer, trânsito)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o propósito de verificar o domínio de aplicação do sistema.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema na área de agricultura (<i>smart agriculture</i>) voltado para capturar dados de distância, peso e saúde de um animal.</p>

Item de Avaliação - Questão 2	
Descrição do Item	O objetivo do sistema está descrito detalhadamente? (Somente visualização de dados; visualização, tomada de decisão e atuação)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar a finalidade para qual o sistema será construído.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados.</p>

Item de Avaliação - Questão 3	
Descrição do Item	O tipo de dado coletado está especificado? (temperatura, umidade, poluição)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar e especificar quais tipos de dados serão tratados no sistema.

Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O format dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes CO2, CO, fumaça.</p>

Item de Avaliação - Questão 4	
Descrição do Item	É possível identificar quem ou o quê coleta os dados? (Sensores, leitores de QR code)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar quais sensores coletam os dados para o sistema.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O format dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados de CO₂ por meio de sensores.</p>

Item de Avaliação - Questão 5	
Descrição do Item	É possível identificar quem ou o quê gerência os dados coletados? (administrador, <i>maker decision</i> , usuários)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar quem administrará os dados do sistema.
Exemplo de Aplicação	

<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois enviar para um servidor que armazena esses dados. Posteriormente, quem tomará decisão mediante os dados coletados.</p>
--	--

Item de Avaliação - Questão 6	
Descrição do Item	É possível identificar quem ou o quê acessa os dados coletados? (coisas, sistemas de software, usuários)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar a finalidade de qual ou quais atores acessam os dados.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Deve-se levar em consideração quais atores terão acesso aos dados. Ex: somente administrador, outros sistemas...</p>

Item de Avaliação - Questão 7	
Descrição do Item	O dispositivo de interface com o usuário que exibe os dados está descrito? (<i>dashboard, smartphone, tablet</i>)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar se o sistema terá uma interface gráfica.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema irá mostrar um painel de gráfico de barras com os dados coletados de CO₂ semanalmente.</p>

Item de Avaliação - Questão 8	
Descrição do Item	É possível identificar quem ou quê visualiza os dados? (coisas, sistemas de software, usuários)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar a quais atores as informações serão exibidas.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema irá mostrar um painel de gráfico de barras com os dados coletados de CO ₂ semanalmente a todos da cidade do Rio de Janeiro.

Item de Avaliação - Questão 9	
Descrição do Item	É possível identificar a fonte de onde os dados são providos ? (cadeiras, mesa, automóveis, casas, prédios)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar a fonte de onde os dados são coletados.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. A fonte de coleta de dados é o ar, devido a concentração de CO ₂ está contida nele.

Item de Avaliação - Questão 10	
Descrição do Item	Os papéis envolvidos com o sistema estão descritos? (coisas, sistemas de software, usuários)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Inconsistência
Objetivo da Questão	Esta questão tem o objetivo de verificar todos os atores envolvidos no sistema.
Exemplo de Aplicação	

<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### A111</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados.</p>
---	---

Item de Avaliação - Questão 11	
Descrição do Item	Existe alguma descrição de cada papel no(s) cenário(s) especificado(s)?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar todos os atores envolvidos no sistema possuem uma descrição formal.
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### A111</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema será encarregado de coletar um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados.</p>

Item de Avaliação - Questão 12	
Descrição do Item	É possível identificar quem ou quê realiza as ações no sistema? (coisas, atuadores, pessoas, sistemas de software)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar todos os atores que realizam ações no sistema.
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### A111</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados.</p>

Item de Avaliação - Questão 13	
Descrição do Item	Cada ação dentro do cenário foi descrita com clareza e não contém informações estranhas?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Informação Estranha
Objetivo da Questão	Esta questão tem o objetivo de verificar todos as possíveis ações no cenário.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. O sistema interpreta o dado e exibe ao usuários finais.

Item de Avaliação - Questão 14	
Descrição do Item	Existe alguma sequência de ações no(s) cenário(s) de compreensão confusa?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Ambiguidade
Objetivo da Questão	Esta questão tem o objetivo de verificar todos os atores envolvidos no sistema.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo que depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados.

Item de Avaliação - Questão 15	
Descrição do Item	Os atores descritos no(s) cenário(s) estão consistentes com os atores descritos nos arranjos – IIA 1, ..., IIA9? (coisas, sistemas de software, usuários)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Fato Incorreto
Objetivo da Questão	Esta questão tem o objetivo de verificar a consistência entre Arranjos IoT e cenários.
Exemplo de Aplicação	

<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. O sistema deve conseguir visualizar os dados (exemplo de fato incorreto).</p>
--	--

Item de Avaliação - Questão 16	
Descrição do Item	O(s) cenário(s) busca(m) estar relacionado(s) aos arranjos?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Fato Incorreto
Objetivo da Questão	Esta questão tem o objetivo de verificar a consistência entre cenários.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados.</p>

Item de Avaliação - Questão 17	
Descrição do Item	São evitados advérbios que gerem mais de uma possibilidade de interpretação nos cenários? (<i>provavelmente, possivelmente, supostamente</i>)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Ambiguidade
Objetivo da Questão	Esta questão tem o objetivo de verificar a clareza dos cenários.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Finalmente, os usuários deverão conseguir visualizar os dados.</p>

Item de Avaliação - Questão 18	
Descrição do Item	O(s) cenário(s) busca(m) ser preciso(s)? (apresentando o objetivo e ações do sistema de forma direta e explícita)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Ambiguidade
Objetivo da Questão	Esta questão tem o objetivo de verificar todos a clareza dos cenários.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados.

Item de Avaliação - Questão 19	
Descrição do Item	São utilizados termos de controle (como "if ou se", "go to", "while") para evitar fluxo de ações menos ambíguo?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Ambiguidade
Objetivo da Questão	Esta questão tem o objetivo de verificar todos os pontos de decisão nos cenários.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Os usuários deverão conseguir visualizar os dados. Se os usuários não conseguirem visualizar os dados. Então os usuários devem notificar o administrador do sistema.

Item de Avaliação - Questão 20	
Descrição do Item	Quando é utilizado palavras como "coisas/things", "dados" no cenário, elas têm o mesmo significado em outras partes desse mesmo cenário?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Inconsistência
Objetivo da Questão	Esta questão tem o objetivo de verificar o "sobrecarregamento" da palavra things/coisa
Exemplo de Aplicação	

<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### A111</p> <ul style="list-style-type: none"> - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal. 	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. As coisas utilizadas no sistema devem ser implantadas e testadas antes. Além disso, devem ser definidos quais coisas coletarão os dados. Os usuários deverão conseguir visualizar os dados obtidos pelas coisas.</p>
--	---

Item de Avaliação - Questão 21	
Descrição do Item	É possível identificar “coisas/ things” descritas com determinada função nos arranjos que representa outra função no(s) cenário(s)?
Resultado Esperado	Sim
Se “não”, descreva o tipo de defeito	Fato Incorreto
Objetivo da Questão	Esta questão tem o objetivo de verificar inconsistências entre artefatos diferentes.
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### A111</p> <ul style="list-style-type: none"> - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal. 	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. As coisas utilizadas no sistema devem ser implantadas e testadas antes. Além disso, devem ser definidos quais coisas coletarão os dados. Os usuários deverão conseguir visualizar os dados obtidos pelas coisas. Os arranjos definem as coisas que serão utilizadas como tags, qr code e sensores.</p>

Item de Avaliação - Questão 22	
Descrição do Item	Os fluxos principais e/ou alternativos e/ou de exceção estão descritos?
Resultado Esperado	Sim
Se “não”, descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar os fluxos nos cenários.
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### A111</p> <ul style="list-style-type: none"> - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal. 	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Caso haja perda de conexão com a internet, os sensores devem continuar coletando os dados. Quando houver reestabelecimento de conexão com a internet, os dados coletados (offline) são enviados para o servidor (exemplo de fluxo alternativo).</p>

Item de Avaliação - Questão 23	
Descrição do Item	A especificação do cenário identifica o arranjo do ID correspondente? (AII1, AII2, ..., AII9)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Inconsistência
Objetivo da Questão	Esta questão tem o objetivo de verificar os cenários e arranjos
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### AII1</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	

Item de Avaliação - Questão 24	
Descrição do Item	É possível identificar o contexto específico no qual o sistema está inserido? (quarto inteligente, estufa IoTizada, veículo autônomo, assistência em saúde smart)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Fato Incorreto
Objetivo da Questão	Esta questão tem o objetivo de verificar o contexto específico do sistema.
Exemplo de Aplicação	
<p># Arranjos</p> <p>## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition)</p> <p>### AII1</p> <p>- Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização).</p> <p>**Explicitar quais devem ser os dados e quais sensores devem ser utilizados**</p> <p>- Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados.</p> <p>**Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados**</p> <p>- O agricultor entra no painel de visualização de informações.</p> <p>- **O formato dos dados ainda deve ser decidido. **</p> <p>- O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</p>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados.</p>

Item de Avaliação - Questão 25	
Descrição do Item	As limitações do ambiente estão descritas? (ex: falta de estrutura de conectividade, falta de estrutura de hardware, infraestrutura inadequada)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar os principais problemas na infraestrutura do ambiente.
Exemplo de Aplicação	

<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AI11 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Caso haja perda de conexão com a internet, os sensores devem continuar coletando os dados. Quando houver reestabelecimento de conexão com a internet, os dados coletados (offline) são enviados para o servidor (exemplo de fluxo alternativo).</p>
--	---

Item de Avaliação - Questão 26	
Descrição do Item	As tecnologias associadas aos objetos do sistema estão descritas? (<i>smart phones, smart watches, wearables</i>)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar os hardwares utilizados pelo sistema.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AI11 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo por meio de sensores, e depois envia para um servidor que armazena esses dados. Caso haja perda de conexão com a internet, os sensores devem continuar coletando os dados. Quando houver reestabelecimento de conexão com a internet, os dados coletados (offline) são enviados para o servidor (exemplo de fluxo alternativo).</p>

Item de Avaliação - Questão 27	
Descrição do Item	Os eventos que o sistema possui foram identificados? (ex: ligar/desligar um objeto, enviar dados)
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar os possíveis eventos dentro dos cenários.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AI11 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Finalmente, o sistema exibe os dados para os usuários.</p>

Item de Avaliação - Questão 28	
Descrição do Item	O tipo de tecnologia de comunicação que o sistema utiliza está descrito nos cenários? (<i>bluetooth, intranet, internet...</i>)
Resultado Esperado	Sim
Se “não”, descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar os tipos de comunicação do sistema.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Caso haja perda de conexão com a internet, os sensores devem continuar coletando os dados. Quando houver reestabelecimento de conexão com a internet, os dados coletados (<i>offline</i>) são enviados para o servidor.</p>

Item de Avaliação - Questão 29	
Descrição do Item	A tecnologia de comunicação proposta atende as especificações geográficas/físicas do sistema? (larga, média ou pequena escala)
Resultado Esperado	Sim
Se “não”, descreva o tipo de defeito	Fato Incorreto
Objetivo da Questão	Esta questão tem o objetivo de verificar o tipo de comunicação do sistema.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### AIII - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. O sistema o=possuirá uma conexão WLAN. Caso haja perda de conexão com a internet, os sensores devem continuar coletando os dados. Quando houver reestabelecimento de conexão com a internet, os dados coletados (<i>offline</i>) são enviados para o servidor.</p>

Item de Avaliação - Questão 30	
Descrição do Item	É possível identificar no(s) cenário(s) como o sistema reage de acordo com as alterações ocorridas no ambiente?
Resultado Esperado	Sim
Se “não”, descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar como o sistema se adapta com mudanças.
Exemplo de Aplicação	

<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. Caso haja perda de conexão com a internet (mudança de estado), os sensores devem continuar coletando os dados. Quando houver reestabelecimento de conexão com a internet, os dados coletados (offline) são enviados para o servidor (mudança de estado novamente).</p>
--	--

Item de Avaliação - Questão 31	
Descrição do Item	As interações do sistema com o ambiente estão representadas no(s) cenário(s)?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar todas as interações que ocorrem no sistema/cenários.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. As informações deverão ser exibidas aos usuários semanalmente.</p>

Item de Avaliação - Questão 32	
Descrição do Item	É possível identificar a interação entre atores?
Resultado Esperado	Sim
Se "não", descreva o tipo de defeito	Omissão
Objetivo da Questão	Esta questão tem o objetivo de verificar a interação entre os atores.
Exemplo de Aplicação	
<pre># Arranjos ## Cenário 1 - Agricultor visualiza informações dos animais (IIA-1: Data exhibition) ### A111 - Os sensores implantados nos animais coletam dados referentes ao ciclo de vida (alimentação e localização). **Explicitar quais devem ser os dados e quais sensores devem ser utilizados** - Os dados coletados pelos sensores no animal são enviados para o sistema e tratados para permitir que sejam visualizados. **Explicitar qual deve ser o tratamento dos dados e onde os dados devem ser visualizados** - O agricultor entra no painel de visualização de informações. - **O formato dos dados ainda deve ser decidido. ** - O sistema disponibiliza a visualização dos dados, que permite que o agricultor verifique se existe necessidade de tomar medidas quanto a situação do animal.</pre>	<p>Um sistema para visualização da poluição de uma parte da cidade do Rio de Janeiro. O sistema coleta um conjunto de dados referentes ao seu objetivo e depois envia para um servidor que armazena esses dados. As informações deverão ser exibidas aos usuários semanalmente.</p>

Apêndice E – Questionário Pós-inspeção

Questionário pós-avaliação da técnica SCENARI_{OT}CHECK

Participante:

1. **Como você classifica o grau de dificuldade da aplicação do SCENARI_{OT}CHECK?**
 - () Muito Fácil
 - () Fácil
 - () Mediano
 - () Difícil
 - () Muito Difícil
2. **Você considera que foi fácil aprender a utilizar a técnica SCENARI_{OT}CHECK?**
 - () Muito Fácil
 - () Fácil
 - () Mediano
 - () Difícil
 - () Muito Difícil
3. **As perguntas da SCENARI_{OT}CHECK foram fáceis de compreender?**
 - () Muito Fácil
 - () Fácil
 - () Mediano
 - () Difícil
 - () Muito Difícil
4. **Você considera que as perguntas da técnica estavam consistentes com o que a técnica estava inspecionando?**
 - () Sim, as perguntas conseguiram tiveram uma boa cobertura.
 - () Não, alguns itens a técnica não teve uma boa cobertura.
5. **Na sua opinião, quais aspectos da técnica tornam sua aplicação fácil/difícil de usar?**
6. **Você identificou os defeitos utilizando somente as perguntas de avaliação da SCENARI_{OT}CHECK?**
 - () Sim. Utilizei apenas as perguntas do SCENARI_{OT}CHECK para me guiar a detectar defeitos no artefato.
 - () Não. Eu utilizei meu conhecimento sobre o assunto em questão.
7. **A técnica SCENARI_{OT}CHECK permitiu você a detectar defeitos mais rápidos que se você fizesse a inspeção sem utilizar técnica alguma?**

- Sim, pelo fato da técnica SCENARI_{OT}CHECK me guiar a questões específicas nos cenários.
- Não.

8. Como o SCENARI_{OT}CHECK auxiliou você a identificar defeitos nos cenários?

- Ruim. O **SCENARI_{OT}CHECK** atuou como um obstáculo. O meu desempenho teria sido melhor se eu não o tivesse utilizado.
- Neutro. Acho que encontraria os mesmos defeitos caso não tivesse utilizado o **SCENARI_{OT}CHECK**.
- Bom. O **SCENARI_{OT}CHECK** me auxiliou na detecção de defeitos. Talvez não tivesse detectado alguns defeitos caso não o tivesse utilizado.

9. Existe alguma pergunta de avaliação que não foi compreendido? Se sim, identifique-o.

10. Você utilizaria a técnica novamente em inspeções futuras?

- Sim
- Não

11. Você considera que as perguntas da técnica SCENARI_{OT}CHECK inspecionam todos os elementos dos cenários que são descritos? Se “sim” ou “não”, por quê?

12. Na sua opinião, como a técnica SCENARI_{OT}CHECK poderia ser melhorada?

Itens de avaliação para identificar defeitos:

Taxonomia para categorização de defeitos:

Apêndice F – Protocolo do Estudo de Observação

1. IDENTIFICAÇÃO

Título: SCENARI_{oT}CHECK: Uma técnica de inspeção baseada em *checklist* para o artefato de cenários

Área Técnica: Engenharia de Software.

Autor: Bruno Pedraça de Souza e Guilherme Horta Travassos.

Afiliação: COPPE/UFRJ.

Local: Rio de Janeiro.

Data: 03 de junho a 14 de maio.

2. CARACTERIZAÇÃO

Será realizado um estudo de observação aplicado em uma disciplina de engenharia de requisitos do curso de engenharia de software.

Tipo: Estudo de Observação.

Domínio: Engenharia de Software Experimental – Especificação e Inspeção de Requisitos para IoT.

Língua: Português (Brasileiro).

Parceiros: Universidade Federal do Amazonas (ICET/UFAM)

Expectativa de Execução: Primeiro Semestre de 2019

Número Estimado de Repetições: Uma

Glossário de Termos:

- Engenharia de Software (ES);
- Engenharia de Sistemas de Software Contemporâneos (ESSC);
- Engenharia de Software Experimental (ESE);
- Internet das Coisas (IoT);
- A Checklist-based Inspection Technique for the Scenario Artifact (SCENARI_{oT}CHECK);
- Perspective Based Reading technique (PBR);
- Usage Based Reading technique (UBR);
- Defect Based Reading technique (DBR);

3. INTRODUÇÃO

Nos últimos anos, estamos vivenciando uma nova era em que máquinas e sistemas estão se tornando cada vez mais independentes, sem precisar do auxílio de humanos para seu funcionamento. Essa “transição” de novas abordagens de sistemas autônomos estão sendo chamadas de quarta revolução industrial pelos pesquisadores e profissionais (LIAO et al, 2017). Em meio a esta mudança, a maneira de interação dos usuários com sistemas está mudando, fazendo assim com que os sistemas se tornem autônomos. E conseqüentemente, tendo o objetivo de facilitar a vida de seus usuários finais. Esses sistemas são chamados de sistemas de software contemporâneos (conhecido pela sigla SSCs). SSCs são todos os novos tipos de sistemas, que de alguma forma são sistemas parciais ou totalmente autônomos. Pode-se classificar esse sistema, como: sistemas sensíveis ao contexto, sistemas baseados em internet das coisas (IoT), sistemas ubíquos e sistemas ciber-físicos (MOTTA et al, 2018).

Com as preocupações dessa “nova revolução industrial”, pesquisadores e profissionais têm pensado em como construir e garantir a qualidade desses novos sistemas (MIORANDI ET AL, 2012) (LIAO ET AL, 2017). Nesse contexto, construir e avaliar sistemas de software contemporâneos não é uma tarefa trivial. Atividades de Engenharia de Software (ES) precisam

estar alinhadas para que o produto final gerado esteja em conformidade com o propósito do sistema construído. Por essa razão, há questionamentos se abordagens, que foram propostas anteriormente, têm a mesma eficácia e eficiência em construir sistemas de software contemporâneos.

A engenharia de requisitos se torna uma preocupação para construção de SSC, devido as abordagens terem sido propostas para construção de sistemas convencionais (mobile e web) e não sistemas contemporâneos. Neste trabalho, iremos avaliar duas abordagens propostas para apoiar a construção destes sistemas. Estas abordagens são aplicadas nas fases iniciais da construção de sistemas de software contemporâneos. Com o intuito de realizar uma especificação de requisitos mais precisa, e depois passando por uma verificação para garantir a conformidade destas especificações.

4. DEFINIÇÃO DO ESTUDO EXPERIMENTAL

- **Objeto de Estudo:** SCENARI_{IoT}CHECK.
- **Objetivo Global:** Este trabalho tem o propósito de realizar um estudo de observação com estudantes de graduação em engenharia de software para aprimorar a utilização da técnica SCENARI_{IoT}CHECK a respeito de verificar artefatos de requisitos produzidos pela técnica SCENARI_{IoT} de descrição de cenários para sistemas de software IoT.
- **Objetivos Específicos**
 - Analisar: SCENARI_{IoT}CHECK - Uma técnica de inspeção baseada em *checklist* para o artefato de cenários
 - Com o propósito de: compreender
 - Em relação: a eficácia, eficiência e percepção do inspetor
 - Do ponto de vista: dos inspetores
 - No Contexto: Uma inspeção do documento de cenários baseado em sistemas de software IoT por estudantes de graduação.
- **Contexto**
 - O contexto desta pesquisa está em engenharia de requisitos e internet das coisas. Nosso contexto está em comparar duas técnicas de inspeção de software a SCENARI_{IoT}CHECK e *ad hoc*. A aplicação da *ad hoc* está relacionada a aplicação do conhecimento do inspetor sem auxílio de uma ferramenta ou técnica. A SCENARI_{IoT}CHECK apoia o inspetor com um conjunto de perguntas para guiá-lo a detectar os defeitos contidos no documento. SCENARI_{IoT} está relacionada a descrição de cenários IoT usando uma serie de arranjos para guiar os inspetores na descrição dos cenários.
- **Questões e Métricas**
 - ***O uso da técnica de inspeção SCENARI_{IoT}CHECK é mais eficaz em encontrar defeitos que da técnica ad-hoc?***

Esta questão tem como objetivo verificar, no contexto do *projeto de CSS*, se a eficácia utilizando a técnica SCENARI_{IoT}CHECK é melhor do que a técnica ad-hoc.
 - ***O uso da técnica de inspeção SCENARI_{IoT}CHECK é mais eficiente em encontrar defeitos que da técnica ad-hoc?***

Esta questão tem como objetivo verificar, no contexto do *projeto de CSS*, se a eficiência utilizando a técnica SCENARI_{IoT}CHECK é melhor do que a técnica ad-hoc.
 - ***Qual a percepção do usuário em relação a ter utilizado a técnica SCENARI_{IoT}CHECK?***

Esta questão tem como objetivo verificar, no contexto do *projeto de CSS*, quais dificuldades os inspetores tiveram ao utilizar a técnica de inspeção SCENARI_{IoT}CHECK.

- Eficiência: nº total de defeitos encontrados pelo tempo (h);
- Eficácia: nº individual de defeitos encontrados de cada inspetor pelo nº total de defeitos encontrados;
- Questionário: coleta da percepção de cada inspetor;
-
- **Questões que não podem ser respondidas no estudo**

As técnicas de inspeções de requisitos convencionais (PBR, DBR, UBR e entre outros) conseguem ter a mesma eficiência e eficácia em encontrar defeitos que técnicas de inspeções específicas para SSC?
- **Questões em aberto**

Qual seria a capacidade de técnicas convencionais de inspeções de requisitos em encontrar os diferentes tipos de defeitos em SSC?

5. PLANEJAMENTO

- **Formulação de Hipóteses**
 - H₀A: Não há diferença entre a eficácia das técnicas *SCENARI_{OT}CHECK* e *Ad hoc*.
 - H₁A: A técnica *SCENARI_{OT}CHECK* é mais eficaz do que a técnica *Ad hoc* para detecção de defeitos.
 - H₀B: Não há diferença entre a eficiência das técnicas *SCENARI_{OT}CHECK* e *Ad hoc*.
 - H₁B: A técnica *SCENARI_{OT}CHECK* é mais eficiente do que a técnica *Ad hoc* para detecção de defeitos.
- **Seleção de Variáveis**
 - **Dependentes:** Eficiência - ***nº total de defeitos encontrados pelo tempo (h)***; Eficácia: ***nº individual de defeitos encontrados de cada inspetor pelo nº total de defeitos encontrados***;
 - **Independentes:** Experiência dos envolvidos em inspeção; técnicas de inspeções; experiência dos envolvidos em descrição de requisitos.
- **Seleção dos Participantes**
 - Critério de Seleção de Participantes: amostra por conveniência
 - Experiência Necessária: será necessário que os estudantes tenham cursado a disciplina de Engenharia de Software
 - Critério de Seleção de Grupos: serão considerados quatorze participantes aplicando *SCENARI_{OT}* para descrição de cenários e as técnicas *Ad Hoc* e *SCENARI_{OT}CHECK* para inspeção de cenários
 - Técnicas Probabilísticas de Amostragem: não se aplica;
 - Técnicas não Probabilísticas de Amostragem: por conveniência.
- **Recursos**
 - Software: artefato de descrição de cenários (SCENARIO);
 - Hardware: Um notebook para a apresentação do material e treinamento em sala de aula;
 - Questionários: será utilizado um questionário para coleta de dados qualitativos, termo de concordância em participação do estudo e caracterização dos participantes (após terem utilizado a técnica *SCENARI_{OT}CHECK*).
- **Design do Experimento**
 - Objetos: artefato de requisitos baseados em cenários;
 - Medições: eficiência, eficácia e percepção do participante;
 - Técnicas: *SCENARI_{OT}*, *Ad Hoc* e *SCENARI_{OT}CHECK*;

- **Instrumentação**
 - Descrição da Instrumentação: nesta parte da pesquisa, as perguntas foram elaboradas para os participantes quando eles forem responder o questionário. Além disso, nós preparamos o roteiro de tarefas que os participantes devem realizar. Será utilizada duas formas para obtenção/coleta dos dados: questionário e observação direta.
 - Apoio à Análise Quantitativa: software R e SPSS;
 - Apoio à Análise Qualitativa: análise dos questionários;
 - Critérios de Observação: sem intervenção dos pesquisadores;
 - Artefatos (Questionários, Procedimentos, etc): questionário;
- **Mecanismos de Análise**
 - Critérios para Eliminação de *Outliers*: não se aplica;
- **Validade dos Resultados**
 - **Validade Interna:** duas ameaças estão sendo consideradas neste estudo, são elas: i) treinamento e ii) tempo medido. Em relação ao (i), nós tentaremos deixar o treinamento de ambas a técnicas equivalentes. Uma vez que não podemos, em hipótese alguma, favorecer nenhuma técnica. Em relação ao item (ii), nós solicitaremos que os estudantes relatem o tempo de início e fim da inspeção.
 - **Validade Externa:** utilização de estudantes de graduação no estudo de observação. Para mitigar esta ameaça, estaremos utilizando o estudo de Carver et al (2003). Assim, estudantes de graduação podem ser utilizados como profissionais iniciantes na indústria, para executar o estudo. Outra ameaça é o estudo ser realizado em ambiente acadêmico. Nós estamos propondo um problema real (relacionado a **agropecuária**), visto que ainda é considerado um problema onde o estudo ocorrerá.
 - **Validade de Conclusão:** Pelo fato do estudo ter uma amostra pequena, o que torna a conclusão do estudo limitada. Porém, com os resultados do estudo de viabilidade, já temos indícios baseados em evidências para termos um sobre a técnica.
 - **Validade de Construção:**

6. TREINAMENTO

- Definição do Treinamento e Procedimentos
 - Aplicador: pelo pesquisador;
 - Participantes: estudantes de graduação;
- Aulas sobre IoT; engenharia de requisito; treinamento sobre descrição de cenários para sistemas de software IoT; treinamento sobre técnicas de inspeções; treinamento sobre técnicas de inspeções para sistemas de software contemporâneos.

7. PROCEDIMENTOS DE EXECUÇÃO

- Definição de Execução do Estudo Experimental: será realizado um estudo de viabilidade com estudantes de graduação dos cursos de engenharia de software e sistemas de informação. Será aplicado em um minicurso nas férias. Inicialmente, será realizado um minicurso sobre IoT e engenharia de requisitos (especificação e inspeção). Após a aula, os alunos serão convidados a realizarem a atividade de inspeção nos documentos que serão dados a eles. No final, os alunos também serão convidados a responderem um questionário sobre a utilização da técnica SCENARI_{IoT}CHECK. O minicurso será realizado nos dias 15 a 18 de janeiro. Duas horas por aula.
- Artefatos (Instruções, Documentos, etc):

- Termo de Consentimento Livre e Esclarecido: para o respaldo e anonimato dos alunos.
- Caracterização dos participantes: para capturar a experiência dos alunos.
- Questionário: para coletar feedback dos participantes após a utilização da SCENARIoT-CHECK.

8. AVALIAÇÃO DO PLANO

- Objetivos: revisado pelos autores deste plano;
- Participantes: dois pesquisadores;
- Procedimentos de Execução: 5 dias (segundo este plano de execução);
- Artefatos Utilizados: técnicas SCENARIO e SCENARIoT-CHECK;
- Artefatos Gerados (Lições Aprendidas, Sugestões de Modificação do Plano): descrição de cenários para sistemas de software contemporâneos (SCENARIO); documento de registro de defeitos (SCENARIoT-CHECK); e questionário de percepção dos participantes após o uso da técnica;

9. PLANEJAMENTO DE CUSTOS

- Custos do Estudo Experimental
 - Custos de Planejamento
 - Plano em si: não se aplica;
 - Instrumentação: não se aplica;
 - Material de Treinamento: não se aplica;
 - Avaliação do Plano: não se aplica;
 - Custos de Execução
 - Deslocamentos: não se aplica;
 - Treinamentos: será aplicado treinamento das técnicas (de descrição de cenários e inspeção para sistemas de software IoT) para os estudantes;
 - Recursos Humanos: estudantes de graduação e pesquisadores;
 - Recursos Materiais: formulários, computadores, software.
 - Custos de Análise: não se aplica;
 - Custos de Empacotamento: não se aplica;

10. REFERÊNCIAS BIBLIOGRÁFICAS

- Motta, R. C.; de Oliveira, K. M.; Travassos, G. H. **On challenges in engineering IoT software systems**. In: the XXXII Brazilian Symposium, 2018, Sao Carlos. Proceedings of the XXXII Brazilian Symposium on Software Engineering - SBES '18. New York: ACM Press, 2018. p. 42.
- Liao, Y.; Deschamps, F.; Loures, E. F. R.; Ramos, L. F. P. **Past, Present and Future of Industry 4.0 - A Systematic Literature Review and Research Agenda Proposal**. International Journal Of Production Research, Vol. 55, No. 12, 3609–3629, 2017.
- Miorandi, D.; Sicari, S.; De Pellegrini, F.; Chlamtac, I. **Internet of things: Vision, applications and research challenges**. Ad Hoc Networks 10, 7 (2012), 1497-1516. DOI:<https://doi.org/10.1016/j.adhoc.2012.02.016>.

Problema: Monitoramento de Animais

O senhor José (agricultor) deseja semi-automatizar sua fazenda de médio porte com aproximadamente 100 hectares com a finalidade de alcançar melhor controle e gerência dos animais e maior produtividade na realização de suas atividades. Para isso, senhor José quer um sistema IoT que permita visualizar a localização de seus animais na fazenda (gados, porcos e ovelhas). Aspectos relacionados à saúde do animal e ao ambiente em que está inserido também devem ser considerados (por exemplo, temperatura corporal, batimento cardíaco do animal [ECG], alimentação, temperatura e umidade, e qualidade do ar do ambiente) essas informações devem armazenadas para futura consulta e/ou análise. Outro aspecto importante é a detecção de alterações na locomoção que pode indicar morte, lesão e/ou doença e pode ser detectado mediante análise dos dados históricos. Caso algum animal apresente algum problema de saúde ou esteja se afastando da fazenda (quadro de roubo/evasão) o sistema deve enviar notificações através de mensagens de SMS e e-mail contendo a identificação do animal, localização atual e estado de saúde. Os recipientes da comida e água espalhados pela fazenda devem ser monitorados. Uma notificação deve ser enviada caso detecte que seja necessária reposição de comida. Caso seja detectada necessidade de água uma bomba deve ser acionada para reabastecer o repositório. Além disso, todas essas informações devem ser apresentadas através de um painel gráfico (*dashboard*) a ser acessado em qualquer tipo de dispositivo seja *smartphones*, computadores convencionais, etc. Este sistema deverá ser semiautônomo, desta forma, precisando do auxílio do agricultor.

Cenários:

- I. **Sistema identifica e coleta dados (animais, ambientes e repositórios de água e comida) e armazena em um banco de dados para futuras análises;**
- II. **Sistema disponibiliza informações sobre os animais cadastrados (localização, batimento cardíaco e temperatura corporal) para o agricultor através de um *dashboard* (permitir a configuração de parâmetros base);**
- III. **Sistema disponibiliza informações sobre os repositórios de comida e de água (localização e quantidade) para o agricultor através de um *dashboard*;**
- IV. **Sistema exibe em um mapa a localização de cada animal e suas respectivas informações, sinalizando aqueles que apresentarem algum problema;**
- V. **Sistema envia uma mensagem para agricultor sobre problemas no ambiente (ex. temperatura acima de 35, umidade acima de 80, qualidade do ar X);**
- VI. **Sistema envia uma mensagem para agricultor sobre problemas cardíacos em determinado animal (DEFINIR REGRA);**
- VII. **Sistema envia uma mensagem para o agricultor avisando que um animal está se afastando da fazenda;**
- VIII. **Sistema envia uma mensagem para agricultor sobre problemas relacionados a alterações na locomoção (ex. acima de 30 minutos em determinado local);**

- IX. Sistema envia uma mensagem para agricultor indicando que está na hora de repor a alimentação de animais em determinado local (ex. repositório abaixo de 5 quilos);**
- X. Sistema detecta repositório de água abaixo do normal e aciona bomba para reabastecer o repositório (ex. repositório abaixo de 5 litros);**
- XI. Sistema analisa dados semanalmente e notifica o agricultor por e-mail com relatório sobre o estado dos animais (comportamento, últimas localizações, média de batimento cardíaco);**
- XII. Sistema analisa dados semanalmente e notifica o agricultor por e-mail com relatório sobre o consumo de alimento e água (abastecimentos) para melhor adaptar e comprar insumos para a próxima semana;**
- XIII. Sistema verifica se existem muitos animais em determinada área e envia notificação para o agricultor (mais de 10 animais a cada 100 metros);**
- XIV. Sistema analisa estoque de comida disponível (sistema de estoque) em estoque e envia notificação ao agricultor caso quantidade esteja abaixo do predeterminado (ex. 20 unidades para cada tipo de animal).**

Apêndice G - Segunda Versão da SCENARI_{IoT}CHECK

Primeira parte da técnica SCENARI_{IoT}CHECK, esta parte tem o objetivo de verificar se as principais características dos sistemas IoT foram capturados nos cenários.

Nº	Questão	Sim	Não	N/A
1	O domínio geral da aplicação foi estabelecido? (saúde, lazer, trânsito)			
2	O objetivo específico do sistema está descrito? (Somente visualização de dados; visualização, tomada de decisão e atuação)			
3	O tipo de dado coletado está especificado? (temperatura, umidade, poluição)			
4	É possível identificar quem ou o quê coleta os dados? (Sensores, leitores de QR code)			
5	É possível identificar quem ou o quê gerência os dados coletados? (administrador, <i>maker decision</i> , usuários)			
6	É possível identificar quem ou o quê acessa os dados coletados? (coisas, sistemas de software, usuários)			
7	O dispositivo de interface com o usuário que exibe os dados está descrito? (<i>dashboard</i> , <i>smartphone</i> , <i>tablet</i>)			
8	É possível identificar quem ou quê visualiza os dados? (coisas, sistemas de software, usuários)			
9	É possível identificar a fonte de onde os dados são providos ? (cadeiras, mesa, automóveis, casas, prédios)			
10	Os papéis envolvidos com o sistema estão descritos? (coisas, sistemas de software, usuários)			
11	Existe alguma descrição de cada papel no(s) cenário(s) especificado(s)?			
12	É possível identificar quem ou quê realiza as ações no sistema? (coisas, atuadores, pessoas, sistemas de software)			
13	Cada ação dentro do cenário foi descrita com clareza e não contém informações estranhas?			
14	Existe alguma sequência de ações no(s) cenário(s) de compreensão confusa?			
15	Os atores descritos no(s) cenário(s) estão consistentes com os atores descritos nos arranjos – IIA 1, ..., IIA9? (coisas, sistemas de software, usuários)			
16	O(s) cenário(s) busca(m) estar relacionado(s) aos arranjos?			
17	O(s) cenário(s) busca(m) ser preciso(s)? (apresentando o objetivo e ações do sistema de forma direta e explícita)			
18	São evitados advérbios que gerem mais de uma possibilidade de interpretação nos cenários? (<i>provavelmente</i> , <i>possivelmente</i> , <i>supostamente</i>)			
19	São utilizados termos de controle (como "if ou se", "go to", "while") para evitar fluxo de ações menos ambíguo?			
20	Quando é utilizado palavras como "coisas/ <i>things</i> ", "dados" no cenário, elas têm o mesmo significado em outras partes desse mesmo cenário?			
21	É possível identificar "coisas/ <i>things</i> " descritas com determinada função nos arranjos que representa outra função no(s) cenário(s)?			
22	Os fluxos principais e/ou alternativos e/ou de exceção estão descritos?			
23	A especificação do cenário identifica o arranjo do ID correspondente? (AII1, AII2, ..., AII9)			

A segunda parte da técnica **SCENARI_{IoT}CHECK**. Esta parte tem o objetivo de verificar se características específicas dos sistemas IoT foram capturados nos cenários.

Nº		Questão	Sim	Não	N/A
24	Ambiente	É possível identificar o contexto específico no qual o sistema está inserido? (quarto inteligente, estufa iotizada, veículo autônomo, assistência em saúde <i>smart</i>)			
25		As limitações do ambiente estão descritas? (ex: falta de estrutura de conectividade, falta de estrutura de hardware, infraestrutura inadequada)			
26	Coisas	As tecnologias associadas aos objetos do sistema estão descritas? (<i>smart phones, smart watches, wearables</i>)			
27	Comportamento	Os eventos que o sistema possui foram identificados? (ex: ligar/desligar um objeto, enviar dados)			
28	Conectividade	O tipo de tecnologia de comunicação que o sistema utiliza está descrito nos cenários? (<i>bluetooth, intranet, internet...</i>)			
29		A tecnologia de comunicação proposta atende as especificações geográficas/físicas do sistema? (larga, média ou pequena escala)			
30	Inteligência	É possível identificar no(s) cenário(s) como o sistema reage de acordo com as alterações ocorridas no ambiente?			
31	Interatividade	As interações do sistema com o ambiente estão representadas no(s) cenário(s)?			
32		É possível identificar a interação entre atores?			

Apêndice H - Registro de Inspeção – Versão 2

REGISTRO DE INSPEÇÃO – SCENARIO_{OT}CHECK				
Data da inspeção: Responsável pela inspeção: Hora inicial: Hora final: Cenário Inspecionado:				
Nº da questão	Sim	Não	N/A	Descrição
Comentário (s):				

Apêndice I - Questionário Pós-inspeção – Versão 2

Participante:

1. Como você classifica o grau de dificuldade da aplicação do SCENARI_{OT}CHECK?

- Muito Fácil
- Fácil
- Mediano
- Difícil
- Muito Difícil

2. Você considera que foi fácil aprender a utilizar a técnica SCENARI_{OT}CHECK?

- Muito Fácil
- Fácil
- Mediano
- Difícil
- Muito Difícil

3. Foi fácil ganhar habilidade no uso da SCENARI_{OT}CHECK?

- Muito Fácil
- Fácil
- Mediano
- Difícil
- Muito Difícil

4. É fácil lembrar como utilizar a SCENARI_{OT}CHECK para realizar uma inspeção de requisitos IoT?

- Muito Fácil
- Fácil
- Mediano
- Difícil
- Muito Difícil

5. A técnica SCENARI_{OT}CHECK permitiu você detectar defeitos mais rápido do que se você fizesse a inspeção sem usar a técnica?

- Concordo Totalmente
- Concordo Parcialmente
- Mediano
- Discordo Parcialmente
- Discordo Totalmente

6. Usar a SCENARI_{OT}CHECK melhorou o meu desempenho na inspeção (acredito ter encontrado um número maior de defeitos do que encontraria sem utilizar a SCENARI_{OT}CHECK)?

- Concordo Totalmente
- Concordo Parcialmente
- Mediano
- Discordo Parcialmente
- Discordo Totalmente

7. As perguntas da SCENARI_{OT}CHECK foram fáceis de compreender?

- Muito Fácil
- Fácil
- Mediano
- Difícil
- Muito Difícil

8. Você considera que as perguntas da técnica estavam consistentes com o que a técnica estava inspecionando?

- Sim, as perguntas conseguiram tiveram uma boa cobertura.
- Não, alguns itens a técnica não teve uma boa cobertura.

9. Na sua opinião, quais aspectos da técnica tornam sua aplicação fácil/difícil de usar?

10. Você identificou os defeitos utilizando somente as perguntas de avaliação da SCENARI_{OT}CHECK?

- Sim. Utilizei apenas as perguntas do SCENARI_{OT}CHECK para me guiar a detectar defeitos no artefato.
- Não. Eu utilizei meu conhecimento sobre o assunto em questão.

11. A técnica SCENARI_{OT}CHECK permitiu você a detectar defeitos mais rápidos que se você fizesse a inspeção sem utilizar técnica alguma?

- Sim, pelo fato da técnica SCENARI_{OT}CHECK me guiar a questões específicas nos cenários.
- Não.

12. Como o SCENARI_{OT}CHECK auxiliou você a identificar defeitos nos cenários?

- Ruim. O SCENARI_{OT}CHECK atuou como um obstáculo. O meu desempenho teria sido melhor se eu não o tivesse utilizado.
- Neutro. Acho que encontraria os mesmos defeitos caso não tivesse utilizado o SCENARI_{OT}CHECK.
- Bom. O SCENARI_{OT}CHECK me auxiliou na detecção de defeitos. Talvez não tivesse detectado alguns defeitos caso não o tivesse utilizado.

13. Existe alguma pergunta de avaliação que não foi compreendido? Se sim, identifique-o.

14. Você utilizaria a técnica novamente em inspeções futuras?

Sim

Não

15. Você considera que as perguntas da técnica SCENARI_{OT}CHECK inspecionam todos os elementos dos cenários que são descritos? Se “sim” ou “não”, por quê?

16. Na sua opinião, como a técnica SCENARI_{OT}CHECK poderia ser melhorada?

- Itens de avaliação para identificar defeitos:

- Taxonomia para categorização de defeitos:

17. Registre quaisquer comentários que julgar pertinente a Técnica SCENARI_{OT}CHECK.

Apêndice J - Termo de Consentimento

Pesquisa: "**SCENARI_{or}CHECK: Uma técnica de leitura baseada em checklist para verificação de cenários em IoT**"

Prezado (a) participante,

Como parte de uma pesquisa do Programa de Pós-Graduação em Engenharia de Sistemas e Computação (PESC/COPPE) da Universidade Federal do Rio de Janeiro (UFRJ), **pedimos gentilmente sua participação nesta pesquisa**, assim para fazermos um estudo sobre duas técnicas de (especificação e inspeção) de requisitos para sistemas de software contemporâneos. Com sua ajuda, o resultado desta pesquisa irá contribuir para aumentar a qualidade de ambas as técnicas propostas, e consequentemente, dos sistemas contemporâneos. **Sua participação não é obrigatória nesta pesquisa.**

1) Procedimento

Serão avaliadas duas técnicas (especificação e inspeção) requisitos. Vale lembrar que você não está sendo avaliado, e sim, a técnica. Após coletar os dados, **seu nome será removido dos mesmos e não será utilizado em nenhum momento durante a análise ou apresentação dos resultados**. Para participar deste estudo peço a sua colaboração em: permitir que os dados resultantes da sua avaliação sejam estudados, e responder um questionário sobre a técnica e um sobre a caracterização.

2) Tratamento de possíveis riscos e desconfortos

Durante a coleta de dados, garantiremos **sua privacidade e seu anonimato**. Os dados coletados durante o **estudo destinam-se estritamente as atividades de pesquisa** relacionadas às técnicas que estão sendo estudadas.

3) Benefícios e Custos

Este estudo contribuirá com resultados importantes para a pesquisa de um modo geral nas áreas de Engenharia de Software e Qualidade de Software. Você não terá nenhum gasto ou ônus com a sua participação no estudo, e também não receberá qualquer espécie de reembolso ou gratificação devido à sua participação **na pesquisa**.

4) Confidencialidade da Pesquisa

Toda informação coletada neste estudo é confidencial e seu nome não será identificado de modo algum, a não ser em caso de autorização explícita para esse fim.

5) Participação

Sua participação neste estudo é muito importante e voluntária. Você tem o direito de não querer participar ou de sair deste estudo a qualquer momento, sem penalidades.

Os pesquisadores responsáveis pelo estudo poderão fornecer qualquer esclarecimento sobre o mesmo, assim como tirar dúvidas, bastando entrar em contato pelos seguintes e-mails:

Aluno: Bruno Pedraça de Souza (COPPE/ UFRJ): bpsouza@cos.ufrj.br

Orientador: Guilherme Horta Travassos (COPPE/ UFRJ): ght@cos.ufrj.br

6) Declaração de Consentimento

Li ou alguém leu para mim as informações contidas neste documento antes de assinar este termo de consentimento. Declaro que toda a linguagem técnica utilizada na descrição deste estudo de pesquisa foi explicada satisfatoriamente e que recebi respostas para todas as minhas dúvidas. Compreendo que sou livre para me retirar do estudo em qualquer momento, sem qualquer penalidade. Declaro ter mais de 18 anos e dou meu consentimento de livre e espontânea vontade para participar deste estudo:

Participante	Pesquisador
Nome: _____	Nome: Bruno Pedraça de Souza
Assinatura: _____	Assinatura: _____

Apêndice K - Caracterização do Participante

Pesquisa: "SCENARI_{OT}CHECK: Uma técnica de leitura baseada em *checklist* para verificação de cenários em IoT"

Participante		Período	
Instituição			

Prezado (a) Senhor (a),

O formulário abaixo será utilizado para compreender seu grau de familiaridade com os diversos aspectos relacionados à especificação e inspeção de requisitos de software e internet das coisas.

A informação coletada será tratada confidencialmente.

- **Conhecimento sobre especificação de requisitos de software**

1. Em relação ao grau do seu conhecimento sobre especificação de requisitos de software, marque a alternativa que melhor se aplica a sua resposta: **Conhecimento sobre especificação de requisitos de software baseado em cenários?**

() Nenhuma

() Tenho noções de especificação/descrição de cenários adquiridas por meio de leituras/palestras

() Estudei especificação/descrição de cenários em aula

2. Em relação ao grau do seu conhecimento sobre especificação de requisitos de software, marque a alternativa que melhor se aplica a sua resposta: **Com que frequência você aplica especificação de requisitos de software?**

() Nenhuma

() Raramente

() Usei apenas uma vez, porém não lembro como se utiliza

() Uso frequentemente especificação de requisitos

Você já realizou/participou de alguma descrição de requisitos baseados em cenários?

() Sim, quantas vezes você já realizou/participou? _____ () Não

- **Conhecimento sobre revisão de requisitos de software**

3. Em relação ao grau do seu conhecimento sobre revisão de requisitos de software, marque a alternativa que melhor se aplica a sua resposta: **Conhecimento sobre inspeção de requisitos de software?**

() Nenhuma

() Tenho noções de inspeções de requisitos de software adquiridas por meio de leituras/palestras

Estudei inspeções de requisitos de software em aula

4. Em relação ao grau do seu conhecimento sobre especificação de requisitos de software, marque a alternativa que melhor se aplica a sua resposta: **Com que frequência você aplica inspeções de requisitos de software?**

Nenhuma

Raramente

Usei apenas uma vez, porém não lembro como se utiliza

Uso frequentemente inspeções de requisitos software

Você já realizou/participou de alguma inspeção de requisitos de software?

Sim, quantas vezes você já realizou/participou? _____ Não

5. Em relação ao grau do seu conhecimento sistemas de software IoT, marque a alternativa que melhor se aplica a sua resposta: **Conhecimento sobre sistemas de software IoT?**

Nenhum

Tenho noções de sistemas de software IoT adquiridos por meio de leituras/palestras

Estudei sistemas de software IoT em aula

6. Você já desenvolveu sistemas de software IoT?

Sim, quantos? _____

Não

Apêndice L – Categorias das Questões do Checklist

	Questão	Tipo de Defeito	Relação entre perguntas
Descrição Geral do Sistema	O domínio geral da aplicação foi estabelecido? (saúde, lazer, trânsito)	Omissão	-
	O objetivo específico do sistema está descrito? (Somente visualização de dados; visualização, tomada de decisão e atuação)	Omissão	Relacionada com Qº 01
	O tipo de dado coletado está especificado? (temperatura, umidade, poluição)	Omissão	-
	É possível identificar quem ou o quê coleta os dados? (Sensores, leitores de QR code)	Omissão	Relacionada com Qº 03
	É possível identificar quem ou o quê gerência os dados coletados? (administrador, <i>maker decision</i> , usuários)	Omissão	-
	É possível identificar quem ou o quê acessa os dados coletados? (coisas, sistemas de software, usuários)	Omissão	Relacionada com Qº 05
	O dispositivo de interface com o usuário que exibe os dados está descrito? (<i>dashboard</i> , <i>smartphone</i> , <i>tablet</i>)	Omissão	Relacionada com Qº 05 e 06
	É possível identificar quem ou quê visualiza os dados? (coisas, sistemas de software, usuários)	Omissão	Relacionada com Qº 05 e 06
	É possível identificar a fonte de onde os dados são providos ? (cadeiras, mesa, automóveis, casas, prédios)	Omissão	Relacionada com Qº 04
	Os papéis envolvidos com o sistema estão descritos? (coisas, sistemas de software, usuários)	Inconsistência	-
	Existe alguma descrição de cada papel no(s) cenário(s) especificado(s)?	Omissão	Relacionada com Qº 10
	É possível identificar quem ou quê realiza as ações no sistema? (coisas, atuadores, pessoas, sistemas de software)	Omissão	Relacionada com Qº 10
	Cada ação dentro do cenário foi descrita com clareza e não contém informações estranhas?	Informação Estranha	Relacionada com Qº 12
	Existe alguma sequência de ações no(s) cenário(s) de compreensão confusa?	Ambiguidade	Relacionada com Qº 11
	Os atores descritos no(s) cenário(s) estão consistentes com os atores descritos nos arranjos – IIA 1, ..., IIA9? (coisas, sistemas de software, usuários)	Fato incorreto	Relacionada com Qº 11
	O(s) cenário(s) busca(m) estar relacionado(s) aos arranjos?	Fato incorreto	-
	O(s) cenário(s) busca(m) ser preciso(s)? (apresentando o objetivo e ações do sistema de forma direta e explícita)	Ambiguidade	Relacionada com Qº 02
	São evitados advérbios que gerem mais de uma possibilidade de interpretação nos cenários? (<i>provavelmente</i> , <i>possivelmente</i> , <i>supostamente</i>)	Ambiguidade	Relacionada com Qº 15
	São utilizados termos de controle (como "if ou se", "go to", "while") para evitar fluxo de ações menos ambíguo?	Ambiguidade	Relacionada com Qº 15
	Quando é utilizado palavras como "coisas/ <i>things</i> ", "dados" no cenário, elas têm o mesmo significado em outras partes desse mesmo cenário?	Inconsistência	Relacionada com Qº 10 e 11
	É possível identificar "coisas/ <i>things</i> " descritas com determinada função nos arranjos que representa outra função no(s) cenário(s)?	Fato incorreto	Relacionada com Qº 10
	Os fluxos principais e/ou alternativos e/ou de exceção estão descritos?	Omissão	-
	A especificação do cenário identifica o arranjo do ID correspondente? (AII1, AII2, ..., AII9)	Inconsistência	-

Questão		Tipo de Defeito	Relação entre perguntas
Ambiente	É possível identificar o contexto específico no qual o sistema está inserido? (quarto inteligente, estufa iotizada, veículo autônomo, assistência em saúde <i>smart</i>)	Fato incorreto	Relacionada com Q ⁹ 01
	As limitações do ambiente estão descritas? (ex: falta de estrutura de conectividade, falta de estrutura de hardware, infraestrutura inadequada)	Omissão	-
Coisas	As tecnologias associadas aos objetos do sistema estão descritas? (<i>smart phones, smart watches, wearables</i>)	Omissão	Relacionada com Q ⁹ 04, 06 e 08
Comportamento	Os eventos que o sistema possui foram identificados? (ex: ligar/desligar um objeto, enviar dados)	Omissão	-
Conectividade	O tipo de tecnologia de comunicação que o sistema utiliza está descrito nos cenários? (<i>bluetooth, intranet, internet...</i>)	Omissão	-
	A tecnologia de comunicação proposta atende as especificações geográficas/físicas do sistema? (larga, média ou pequena escala)	Fato incorreto	-
Inteligência	É possível identificar no(s) cenário(s) como o sistema reage de acordo com as alterações ocorridas no ambiente?	Omissão	Relacionada com Q ⁹ 12
Interatividade	As interações do sistema com o ambiente estão representadas no(s) cenário(s)?	Fato incorreto	-
	É possível identificar a interação entre atores?	Omissão	Relacionada com Q ⁹ 12

Apêndice M – Template para Descrição de Cenários IoT

Nome do Projeto	<i>[nome do projeto]</i>		
Responsável pelo Documento	<i>[responsável pela criação do documento]</i>	Data de Criação	<i>[dd/mm/aa]</i>
Objetivo do Sistema	<i>[descrever o objetivo deste sistema em termos de propósito e finalidade]</i>		
Domínio do Sistema	<i>[descrever o domínio do sistema, como saúde, lazer, trânsito e assim por diante]</i>		
Atores	<i>[descrever os atores do sistema como usuários; coisas; sistemas de software;]</i>		
Tipos de Dados Coletados	<i>[descrever tipos de dados coletados por sensores, como por exemplo, temperatura; unidade; tempo; luminosidade; e assim por diante]</i>		
ID do Cenário	<i>SC[id]</i>	Título	<i>[título do cenário]</i>
Arranjo	<i>[IIA-01, ..., IIA-09] e nome do arranjo</i>		
Atores	<i>Usuários: [descrevem os usuários como: usuário final, animais...]</i>		
	<i>Coisas: [descreva as coisas com seus sensores, atuadores, wearables ...]</i>		
	<i>Sistemas de software: [descreva os sistemas de software]</i>		
Passos (Todas as etapas do sistema devem ser descritas detalhadamente)	Sequência de Interação		
	<i>[FLUXO PRINCIPAL - descreva as etapas do cenário usando os atores descritos anteriormente e suas respectivas interações no arranjo. Coleta e tratamento de dados também devem ser contemplados. Lembre-se, os cenários precisam ser objetivos e claramente compreendidos].</i>		
	<i>[FLUXO ALTERNATIVO - descreva os fluxos alternativos que o cenário possui].</i>		
	<i>[FLUXO DE EXCEÇÃO - descreva os fluxos de exceção que o cenário possui].</i>		
	<i>[REGRAS DE NEGÓCIOS - descreva as regras de negócio pertinentes ao cenário].</i>		
Ambiente	<i>[descrever o ambiente e suas limitações “O ambiente é o lugar onde as coisas estão, as ações acontecem, os eventos ocorrem e os usuários estão.”]</i>		
Conectividade	<i>[descreva o tipo de conectividade necessária (ex: wired, wirelles, etc.), o tamanho da cobertura da rede e seu tipo (ex: low scale - PAN; média escala - MAN; alta escala – WAN), etc.]</i>		