

UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Uma abordagem para o desenvolvimento de aplicações para a internet das coisas utilizando conceitos de sistema-de-sistemas

Danillo Ferreira dos Reis

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional (PPG-C²MC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Danillo Ferreira dos Reis

Uma abordagem para o desenvolvimento de aplicações
para a internet das coisas utilizando conceitos de
sistema-de-sistemas

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Rosana Teresinha Vaccare Braga

USP – São Carlos
Fevereiro de 2019

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

R375a Reis, Danillo Ferreira dos
 Uma abordagem para o desenvolvimento de
aplicações para a internet das coisas utilizando
conceitos de sistema-de-sistemas / Danillo Ferreira
dos Reis; orientadora Rosana Teresinha Vaccare
Braga. -- São Carlos, 2019.
 99 p.

Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2019.

1. Internet das Coisas. 2. Sistema-de-Sistemas.
3. Arquitetura Orientada a Serviços. I. Braga,
Rosana Teresinha Vaccare, orient. II. Título.

Danillo Ferreira dos Reis

An approach for developing applications to the internet of things using system-of-systems concepts

Master dissertation submitted to the Institute of Mathematics and Computer Sciences – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Rosana Teresinha Vaccare Braga

USP – São Carlos
February 2019

Dedico este trabalho aos meus familiares, em especial ao meu pai, minha mãe, meu irmão e minha sobrinha, que sempre estiveram comigo em todas as situações e não mediram esforços para me apoiarem e darem suporte durante esta caminhada.

AGRADECIMENTOS

Acredito que agradecer seja fundamental na vida das pessoas. É importante ser grato nos bons e maus momentos da vida e, além disso, ter a sabedoria de tirar proveito do que foi importante nos momentos difíceis. Desses pouco mais de três anos vividos na cidade de São Carlos/SP, houveram diversos bons momentos que guardarei para sempre em minha memória. Também houveram momentos difíceis, que me fizeram cair e querer desistir, porém me fizeram aprender, crescer, evoluir e ter forças para encarar os obstáculos da vida. É como diz a frase de Johnny Deep: “Você nunca sabe a força que tem, até que a sua única alternativa é ser forte”.

Dessa forma, quero agradecer, primeiramente, aos meus amados pais Lázaro e Aparecida, ao meu irmão Gabriel e a minha sobrinha Gabrielle (cabritinha) que me deram todo o apoio e carinho necessário para trilhar esse caminho árduo da pós graduação. Sem o apoio deles nada disso seria possível de ser concretizado.

À Profa Dra Rosana Braga, que me orientou com muita sabedoria e paciência, guiando sempre pelos melhores caminhos da pesquisa. A senhora foi mais que uma orientadora, foi uma mãe que sempre entendeu e ajudou a superar alguns problemas ocorridos durante esse processo. Continue essa pessoa incrível que a senhora sempre foi.

Aos familiares (tias, tios, primas, primos e afilhados) que, de uma forma ou de outra, me deram ânimo e inspiração para a conclusão deste trabalho. São muitos os nomes, então não me atrevo a escrever para não esquecer nenhum. Porém todos sabem da importância que tem para mim.

À minha segunda família Zico, Cida, Thayane, Thelles, Igor e Odair Filho que sempre me fizeram sentir em casa fora de casa. Cada incentivo, palavra de carinho e afeto foram importantes para a conclusão deste trabalho.

Às pessoas que também foram importantes para a conclusão desse trabalho, sendo eles: Anniely, Murilo, Denis, Guilherme, João Paulo, Alyne, Lorrany, Tathy, Priscila Rebouças, Arthur, Marco Antônio (Marquim), Raphael, Weider, Wênder, Dyeimys, Francisco Neto, Priscila Gomes, Lorena, Ana Cristina, Camila Brito e outros. Devo ter me esquecido de algumas pessoas, mas saibam que todos foram importantes de alguma forma durante esse processo.

À Profa Dra Ana Carolina Gondim, que me orientou durante a graduação na UFG. Obrigado pelos ensinamentos, conselhos e “catracadas” para que meu trabalho de conclusão de curso fosse possível de ser concluído.

Aos amigos da graduação que também vieram para o caminho da pós-graduação em São

Carlos. Graças à eles, minha adaptação na cidade se tornou bem menos difícil. São eles: Pedro Henrique, João Paulo (Jãum), Renato, Diógenes e Ricardo.

Aos amigos que conheci na pós graduação na USP e também na UFSCar, pelas risadas, conversas nos corredores, conselhos, estudos durante o período de disciplinas, aniversários surpresas, produção de artigo científico, almoços, jantares, lanches, café da tarde na cantina, almoços no R.U., churrascos da pós graduação, churrascos entre amigos, seminários e confraternizações do laboratório, vôleis e, claro, toda a ajuda durante a construção deste projeto. São eles: Brauner, Cristiane, Franciele Vargas, Francielle Mattos, Lívia, Iohan, Silvana, Nilton, Lina, Daniel, Valdemar, Thiago Gottardi, Tiago Volpato, Kamila, Diego Damasceno, Diego, Armando, Stevão, Paulo Júnior (Paulão), Laíza, Alex Guido, Sidgley, Alinne, FCarlos, Willmax, Gisele, pequeno Davi, Misael, Delcio, Ronnie Shida, Yuri, André, Renan Shida, Renan Vinícius (amigo da UFG e também da USP - de campus diferentes), Valéria, Jadson, Jacson, Vítinho, Vitor Barella, Rafael Mantovani, Adam, Lucas e William. É importante ressaltar que os amigos citados no parágrafo anterior também fizeram parte de todas as risadas, conversas, e etc citado no início deste parágrafo! Foi uma imensa honra ter trabalhado e convivido momentos singulares com todos vocês durante esse período.

Aos novos amigos que conheci durante o ano de conclusão deste trabalho, que me proporcionaram momentos de descontração e boas conversas. São eles: João Choma, João Biazotto, Henrique, Thiago Costa e Jorge.

Ao CNPq pelo auxílio financeiro para a realização desta pesquisa.

Ao Instituto de Ciências Matemáticas e de Computação e a Universidade de São Paulo (campus São Carlos), em especial ao Laboratório de Engenharia de Software (LabES), por me proporcionar um ambiente de trabalho digno de um aluno de pós graduação.

A Dona Ana, Aurinha e ao Sandro Colângelo (*in memoriam*), que trabalham na cantina e sempre me recebiam com um sorriso no rosto nas horas do café da manhã e do lanche da tarde. É sempre bom ter pessoas assim no nosso dia a dia.

Enfim, todos foram de extrema importância. Perdoem-me por esquecer alguns nomes, mas saibam que tenho muito apressado por cada pessoa que passou por minha vida.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

*“As invenções são, sobretudo,
o resultado de um trabalho de teimoso.”
(Santos Dumont)*

RESUMO

REIS, D. F. **Uma abordagem para o desenvolvimento de aplicações para a internet das coisas utilizando conceitos de sistema-de-sistemas**. 2019. 99 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Nos últimos anos, a Internet expandiu-se de forma exponencial e tornou-se um sistema de comunicação entre milhões de pessoas. Em paralelo, houve um crescente avanço tecnológico na computação e comunicação. Nessa perspectiva, conceitos associados à “Internet do Futuro” e, em particular, à Internet das Coisas (IoT), têm sido pesquisados e amplamente aplicados em diferentes domínios, tais como transporte e saúde. No entanto, por ser um assunto recente, foi verificado em um mapeamento sistemático que existem lacunas de pesquisa em relação às abordagens de desenvolvimento de aplicações IoT. Por outro lado, foram identificadas semelhanças entre os conceitos de IoT e de Sistemas-de-Sistemas (SoS), que é uma composição de diversos sistemas constituintes (SCs), os quais possuem independência gerencial e operacional. Quando os SCs são integrados para compor um SoS, ocorrem comportamentos emergentes não disponíveis em quaisquer dos SCs individualmente. Em ambientes como cidades inteligentes, diversos sistemas heterogêneos podem se integrar rumo a um SoS, gerando benefícios aos cidadãos. Dessa forma, o objetivo deste trabalho de mestrado foi estabelecer uma abordagem para auxiliar o desenvolvimento de aplicações IoT com base nos conceitos de SoS, realizando a integração entre os diferentes tipos de SCs presentes em um ambiente inteligente por meio da utilização do paradigma orientado a serviços. Foi também apresentada uma prova de conceito em que a abordagem foi utilizada para desenvolver um SoS na área de controle de epidemia causada por mosquitos.

Palavras-chave: Internet das Coisas, Sistema-de-Sistemas, Arquitetura Orientada a Serviço.

ABSTRACT

REIS, D. F. **An approach for developing applications to the internet of things using system-of-systems concepts**. 2019. 99 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2019.

Recently, the Internet has exponentially expanded and became a communication system among millions of people. In parallel, there has been an increasing technological advance in computing and communication. Concepts related to the “Internet of the Future” and, particularly, the Internet of Things (IoT), have been widely investigated and employed in different domains, such as transportation and health. However, there are research gaps with regard to the development approaches of IoT applications, according to a systematic mapping. On the other hand, it has been identified similarities between the concepts of IoT and Systems-of-Systems (SoS), which is a composition of several constituent systems (CSs) with managerial and operational independence. A characteristic of SoS is to allow a behavior that is not available in SCs individually. In the smart cities domain, for instance, heterogeneous systems can be integrated into a SoS, generating benefits for citizens. In this perspective, the objective of this Master’s work is to establish an approach for the development of IoT applications based on the concepts of SoS, realizing the integration between the different types of CSs present in an intelligent environment through the use of the service-oriented paradigm. A proof of concept was carried out, where the approach was employed to develop a SoS for epidemic control.

Keywords: Internet of things, System-of-Systems, Software-Oriented Architecture.

LISTA DE ILUSTRAÇÕES

Figura 1 – Definição de IoT	30
Figura 2 – Interfaces do aplicativo Waze	34
Figura 3 – Lixeira inteligente	35
Figura 4 – Tipos de SoS	38
Figura 5 – SoS de assistência médica	40
Figura 6 – Rover MastCam	40
Figura 7 – Visão simplificada de SOA	41
Figura 8 – Conteúdo da descrição de um serviço	44
Figura 9 – Processo geral de desenvolvimento de aplicações IoT	46
Figura 10 – Metodologia do <i>IDeA</i>	47
Figura 11 – Modelo de domínio IoT	48
Figura 12 – Fases para a realização do mapeamento sistemático	52
Figura 13 – Passos do Mapeamento Sistemático	56
Figura 14 – Processo de seleção dos estudos primários	58
Figura 15 – Abordagem proposta pelo proponente	65
Figura 16 – Exemplos de integração de aplicações IoT e de outros domínios	66
Figura 17 – Representação alto nível utilizando diagrama de classes	69
Figura 18 – Representação alto nível utilizando diagrama de sequência - SoS Proativo	69
Figura 19 – Representação alto nível utilizando diagrama de sequência - SoS Reativo	70
Figura 20 – Modelo conceitual da abordagem proposta na fase de projeto	73
Figura 21 – Cenário ilustrativo do SoS Sai Dengue	81
Figura 22 – Diagrama de caso de uso geral da simulação SoS Sai Dengue	82
Figura 23 – Diagrama de caso de uso da simulação SoS Sai Dengue com as entidades responsáveis	83
Figura 24 – Modelo conceitual da simulação SoS Sai Dengue	84
Figura 25 – Interação entre os principais SCs definidos da simulação	85
Figura 26 – Serviço “ObterDadosSensores”	85
Figura 27 – Serviço “obterInformacaoSensores”	86
Figura 28 – Serviço “receberAlertaSecServPublicos”	86
Figura 29 – Serviço “receberAlertaSaude”	86
Figura 30 – Sistemas envolvidos na simulação	87
Figura 31 – Projetos da implementação da simulação	88
Figura 32 – Consumindo o serviço “obterInformacoesSensores”	90

LISTA DE TABELAS

Tabela 1 – Métodos HTTP	43
Tabela 2 – Papéis dos <i>stakeholders</i> no desenvolvimento de aplicações IoT	46
Tabela 3 – Diferentes <i>Stakeholders</i> no desenvolvimento de aplicações IoT	49
Tabela 4 – Palavras-chave e sinônimos	54
Tabela 5 – String de busca	54
Tabela 6 – Base de dados selecionadas	55
Tabela 7 – Quantidade de trabalhos retornados por base de dados	57
Tabela 8 – Estudos selecionados para análise e extração de dados	59
Tabela 9 – Fases de desenvolvimento dos estudos selecionados	61
Tabela 10 – Papéis dos <i>stakeholders</i> no passo de análise	67
Tabela 11 – Papéis dos <i>stakeholders</i> na fase de projeto	74
Tabela 12 – Papéis dos <i>stakeholders</i> na fase de implementação	75
Tabela 13 – Anotações REST	75
Tabela 14 – Descrição de uso do cenário principal	81
Tabela 15 – Serviços REST implementados na simulação	89

SUMÁRIO

1	INTRODUÇÃO	23
1.1	Contextualização e Motivação	23
1.2	Justificativa	25
1.3	Objetivo e Metodologia	27
1.4	Organização do trabalho	28
2	FUNDAMENTAÇÃO TEÓRICA	29
2.1	Considerações Iniciais	29
2.2	Internet das Coisas	29
2.2.1	<i>Definição</i>	30
2.2.2	<i>Breve Histórico</i>	31
2.2.3	<i>Características da Internet das Coisas</i>	31
2.2.4	<i>Aplicações</i>	32
2.2.4.1	<i>Automação Doméstica</i>	32
2.2.4.2	<i>Indústria Médica e de Cuidados com a Saúde</i>	33
2.2.4.3	<i>Monitoramento de Ambientes</i>	33
2.2.4.4	<i>Transporte e Trânsito</i>	33
2.2.4.5	<i>Reciclagem</i>	34
2.2.4.6	<i>Agricultura e criação de animais</i>	35
2.3	Sistema-de-Sistemas	36
2.3.1	<i>Definições e Características</i>	36
2.3.2	<i>Tipos de SoS</i>	38
2.3.3	<i>Aplicações</i>	39
2.4	Arquitetura Orientada a Serviços	40
2.4.1	<i>Características</i>	41
2.4.2	<i>Web Services</i>	42
2.4.2.1	<i>Estrutura do web service</i>	43
2.4.2.2	<i>Papéis do web service</i>	43
2.5	Trabalhos Relacionados	44
2.6	Considerações Finais	49

3	ABORDAGENS PARA O DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET DAS COISAS - UM MAPEAMENTO SISTEMÁTICO	51
3.1	Considerações Iniciais	51
3.2	Mapeamento Sistemático	52
3.2.1	<i>Planejamento</i>	53
3.2.1.1	<i>Objetivo e questões de pesquisa</i>	53
3.2.1.2	<i>String de busca</i>	54
3.2.1.3	<i>Seleção das bases de dados</i>	54
3.2.1.4	<i>Critérios de inclusão e exclusão</i>	55
3.2.2	<i>Condução</i>	55
3.2.3	<i>Sumarização</i>	57
3.3	Considerações Finais	62
4	UMA ABORDAGEM PARA AUXILIAR O DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET DAS COISAS UTILIZANDO CONCEITOS DE SISTEMA-DE-SISTEMAS	63
4.1	Considerações iniciais	63
4.2	Visão Geral da abordagem	64
4.3	Passo 1: Requisitos	67
4.4	Passo 2: Projeto	70
4.5	Passo 3: Implementação	72
4.6	Discussão	76
4.7	Considerações finais	77
5	UMA PROVA DE CONCEITO DA ABORDAGEM	79
5.1	Considerações iniciais	79
5.2	Descrição do exemplo utilizado	80
5.3	O processo de desenvolvimento	80
5.3.1	<i>Requisitos</i>	81
5.3.2	<i>Projeto</i>	83
5.3.3	<i>Implementação</i>	88
5.4	Funcionamento da simulação	89
5.5	Considerações Finais	90
6	CONCLUSÃO	91
6.1	Visão Geral da Pesquisa e Contribuições	91
6.2	Dificuldades e Limitações	92
6.3	Trabalhos Futuros	92

REFERÊNCIAS 95

INTRODUÇÃO

1.1 Contextualização e Motivação

A Internet teve um grande crescimento nos últimos anos. Na década de 1980, conectar computadores em uma rede era apenas um projeto de pesquisa. Com o crescimento das redes, a Internet tornou-se um sistema de comunicação que alcança milhões de pessoas em todo o mundo. Geralmente, os usuários usam diferentes tecnologias, tais como Linha Digital de Assinante (do inglês, *Digital Subscribe Line* - DSL), fibra óptica, entre outros, para ter acesso à Internet de alta velocidade (COMER, 2016).

Esse crescimento fez com que a Internet se tornasse uma fonte de informações disponível 24 horas por dia, sendo considerada uma entidade viva que sempre está mudando e evoluindo. As redes, com fio e sem fio, estão por toda parte e permitem que a conectividade esteja cada vez mais presente na sociedade, fazendo com que as pessoas estejam “sempre conectadas” (COETZEE; EKSTEEN, 2011). É possível encontrar diferentes serviços por meio do uso da Internet, tais como redes sociais, compartilhamento de músicas e vídeos, realização de compras online, jogos online, e pesquisa de informações.

Com o avanço da tecnologia, conceitos associados a “Internet do Futuro” estão sendo pesquisados e aplicados (COETZEE; EKSTEEN, 2011). Um desses conceitos é a denominada Internet das Coisas (do inglês, *Internet of Things* - IoT), uma revolução tecnológica que representa o futuro da computação e das comunicações (TAN; WANG, 2010). Por meio dela, é possível que objetos do dia a dia, tais como eletrodomésticos, veículos e até roupas, troquem informações entre si, aumentando a qualidade de vida e otimizando processos industriais, além de automatizar tarefas que os humanos devem executar (COSTA; PIRES; DELICATO, 2016).

Isso resulta em diversos benefícios, por exemplo, aumentar o conforto e segurança de escritórios e residências, otimizar cuidados com a saúde, reduzir o consumo de energia elétrica, auxiliar o tráfego no trânsito, dentre outros. Tais benefícios são obtidos por meio do

desenvolvimento de aplicações IoT. Segundo [Costa, Pires e Delicato \(2016\)](#), uma aplicação IoT é definida como “*uma coleção de procedimentos automatizados e dados, integrados com entidades heterogêneas (hardware, software, e pessoal) que interagem entre si e com seu ambiente para atingir objetivos comuns*”.

Com isso, aplicações IoT estão sendo desenvolvidas cada vez mais em diferentes domínios, tais como transporte, saúde e manufatura, trazendo diversos benefícios para o dia a dia das pessoas. Um edifício, por exemplo, pode interagir com seus moradores e também com prédios vizinhos em caso de incêndio para segurança dos residentes; escritórios podem ajustar-se automaticamente por meio do uso de dispositivos inteligentes de acordo com as preferências dos funcionários, minimizando o consumo de energia, ou notificações sobre o fluxo de veículos de acordo com o status da rodovia em tempo real ([PATEL; CASSOU, 2015](#)).

Um outro exemplo é observado na medicina com a existência de uma aplicação que utiliza biosensor para medir o nível de açúcar no sangue por meio da análise da saliva. Essa aplicação torna o diagnóstico do usuário mais preciso e rápido, além de não ter a necessidade de picada de agulha para recolher amostra de sangue, trazendo uma melhor qualidade de vida para pessoas que possuem diabetes ([PACIFICO, 2016](#)). Essas aplicações envolvem a interação de um amplo número de dispositivos diferentes, como sensores e atuadores, interagindo com o seu ambiente físico ([PATEL; CASSOU, 2015](#)).

No entanto, mesmo com os diversos benefícios que a IoT proporciona para a sociedade, ainda existem dificuldades no processo de desenvolvimento de aplicações IoT. Essas geralmente utilizam dispositivos móveis para sua aplicação. Segundo [Aziz, Sheikh e Felemban \(2016\)](#), algumas práticas de Engenharia de Software podem apoiar o desenvolvimento dessas aplicações. No entanto, algumas adaptações nos processos de desenvolvimento de software tradicional devem ser realizadas para apoiar o desenvolvimento de aplicações IoT ([COSTA; PIRES; DELICATO, 2016](#)).

São muitos os desafios encontrados no desenvolvimento de aplicações IoT em diversas fases. Na especificação de modelos de sistema, por exemplo, as limitações encontradas são: i) a heterogeneidade de dispositivos de hardware e componentes de software; ii) a falta de mecanismos que permitam a descrição dos interesses dos *stakeholders*; e iii) a existência de poucos métodos para projetar aplicações IoT ([COSTA; PIRES; DELICATO, 2016](#)).

No mapeamento sistemático apresentado na Seção 3, foi evidenciado que existe uma carência de estudos que propõem abordagens para desenvolvimento de aplicações IoT. Além disso, [Chen et al. \(2015\)](#) discutem outra dificuldade para o desenvolvimento de aplicações IoT, que é o fato das aplicações IoT serem feitas próximas ao sistema operacional. Com isso, os programadores se preocupam com questões de baixo nível da aplicação e também não se atentam a realização da documentação inicial, necessária para o entendimento e manutenção das aplicações ([CHEN et al., 2015](#)).

Isso dificulta o entendimento do código resultante, por não haver representação das aplicações em níveis mais altos de abstração. A engenharia de software pode ajudar a resolver esse problema, por meio de atividades como análise de requisitos e análise de sistemas, que produzem diversos modelos para ajudar no entendimento e futura evolução do sistema. Uma iniciativa foi encontrada no contexto de engenharia de requisitos para IoT (AZIZ; SHEIKH; FELEMBAN, 2016), em que casos de uso são utilizados para especificar o comportamento em aplicações IoT.

1.2 Justificativa

Quando se integram aplicações IoT para formar sistemas maiores, como no exemplo de cidades inteligentes, pode-se notar a semelhança com o conceito de Sistema-de-Sistemas (do inglês *System of Systems* - SoS). SoS, de acordo com Seo *et al.* (2016), é um conjunto de sistemas constituintes (SCs) com independência operacional e gerencial, visando cooperar uns com os outros para atingir um objetivo comum e maior (comportamento emergente).

Segundo Maia *et al.* (2014), é possível integrar aplicações IoT para formar SoS, facilitando a obtenção de comportamentos emergentes entre os SCs. Da mesma forma que em SoS, em IoT existem “coisas” que podem estar envolvidas em relacionamentos complexos por envolver outros sistemas independentes e heterogêneos, como por exemplo em cidades inteligentes. O mapeamento sistemático realizado por Maia e outros investigou a existência de abordagens para permitir a construção de SoS nos quais sistemas baseados em IoT pudessem ser integrados para formar o SoS, os quais são denominados pelos autores como *SoS baseados em IoT*. Os resultados mostraram que os trabalhos encontrados estão em um estágio inicial de desenvolvimento, enfocando no entendimento dos requisitos dos SoS baseados em IoT, bem como em arquiteturas de alto nível para apoiar tais sistemas.

Adicionalmente, há pesquisas propondo plataformas de *middleware* para lidar com heterogeneidade e interoperabilidade dos sistemas. No entanto, há várias oportunidades de pesquisa a serem investigadas no futuro para permitir o desenvolvimento mais efetivo de SoS baseados em IoT. Exceto pelo trabalho de Maia *et al.* (2014), não foram encontrados estudos comparando SoS e IoT especificamente, o que abre a possibilidade de iniciar uma investigação mais aprofundada sobre esse relacionamento. Os resultados dessa investigação podem levar a abordagens que aproveitem as vantagens de SoS no desenvolvimento de aplicações IoT. Além disso, pode-se analisar se as aplicações IoT podem ser consideradas um tipo de aplicação de SoS.

A abordagem de Maia *et al.* (2014) é diferente da que se investigou neste trabalho. Enquanto Maia e colaboradores consideram SoS compostos pela integração de aplicações IoT, neste trabalho investigou-se SoS em que os SCs são as “coisas” no contexto de IoT. Os SCs, por sua vez, são sistemas independentes operacionalmente e gerencialmente. Em particular, almeja-se

facilitar o reúso de SCs, que podem participar de vários SoS diferentes simultaneamente.

Além da análise feita por [Maia et al. \(2014\)](#) sobre SoS e IoT, [Cavalcante et al. \(2016\)](#) realizou um estudo comparativo entre cidades inteligentes (do inglês, *smart cities*) e SoS. De acordo com os autores, as cidades estão evoluindo em ambientes que consistem de diversos sistemas distribuídos em complexos relacionamentos, incluindo a integração e a interação com outros sistemas para fornecer novas funcionalidades. Essas relações levam cidades inteligentes a se tornarem SoS, um conjunto de sistemas (SCs) independentes que colaboram uns com os outros para atingirem um objetivo maior.

É possível encontrar vários SoS em diversos campos em cidades inteligentes, tais como saúde, transporte, energia, monitoramento, gerenciamento de desastres e segurança pública. Uma cidade inteligente pode integrar sistemas independentes e heterogêneos, públicos e privados em diferentes domínios ([CAVALCANTE et al., 2016](#)).

Devido à heterogeneidade de sistemas no âmbito de SoS, umas das preocupações é a forma como os SCs interoperam para que a missão seja atingida. Dessa forma, a Arquitetura Orientada a Serviços (do inglês *Service-Oriented Architecture* - SOA) tem sido adotada para resolver esse problema. SOA tornou-se uma maneira prática de desenvolvimento de software para implementar sistemas distribuídos devido à sua praticidade e baixo acoplamento ([LASKEY et al., 2009](#); [FURTADO et al., 2009](#)).

De acordo com a revisão sistemática realizada por [Vargas, Gottardi e Braga \(2016\)](#) para averiguar abordagens, técnicas e ferramentas que auxiliam a integração de sistemas, foi evidenciado que grande parte dos estudos verificados, 54,16%, utilizam SOA como uma solução arquitetural no contexto de SoS. Além do baixo acoplamento, SOA oferece padronização, independência de plataforma, interfaces bem definidas e suporte a ferramentas que permitem a integração de sistemas legados, ideal para o desenvolvimento de sistemas rumo a um SoS bem sucedido ([LEWIS et al., 2011](#); [FURTADO et al., 2009](#)).

Sendo assim, a realização desta pesquisa partiu da conclusão de que ainda existem poucos métodos que auxiliem o desenvolvimento de aplicações IoT, visto que em um ambiente inteligente, como em uma cidade inteligente, existem diversas aplicações ou sistemas heterogêneos que podem interoperar uns com os outros, formando um SoS. Além da existência de poucos métodos que apoiem o desenvolvimento de aplicações, foi evidenciado, por meio da realização de um mapeamento sistemático executado pelo proponente e pela orientadora deste trabalho, que são poucos os métodos existentes que apoiam todas as fases do processo de desenvolvimento de aplicações IoT. Segundo [Costa, Pires e Delicato \(2016\)](#), cada fase no desenvolvimento de aplicações IoT apresenta uma gama de desafios para os *stakeholders*, pois os problemas enfrentados em cada uma delas devem ser adaptados para o contexto de IoT.

1.3 Objetivo e Metodologia

Este projeto de mestrado teve como principal objetivo estabelecer uma solução que ajude desenvolvedores a elaborar aplicações IoT, visando auxiliar o processo de desenvolvimento, desde as fases iniciais até as fases finais. O objetivo da elaboração deste trabalho surgiu a partir da conclusão da falta de métodos que auxiliem no desenvolvimento de aplicações IoT.

Com isso, a questão de pesquisa que norteou a realização deste trabalho foi averiguar o uso de conceitos de SoS no desenvolvimento de aplicações que envolvem dispositivos IoT, pois devido à existência de diversos SCs (que podem ser sistemas convencionais ou sistemas baseados em “coisas”), comportamentos maiores (comportamento emergente) podem ser alcançados para proporcionar melhor qualidade de vida das pessoas.

Devido à utilização de conceitos de SoS para o desenvolvimento de aplicações IoT, faz parte dessa abordagem a utilização de SOA para integração entre “coisas” e sistemas convencionais em ambientes inteligentes. A disponibilização de recursos por meio de serviços faz com que diversas aplicações possam interagir, independentemente da plataforma adotada.

A abordagem proposta compreende diversos passos direcionando os desenvolvedores na elaboração de aplicações IoT, desde a fase de obtenção dos requisitos de desenvolvimento até a fase de implementação, utilizando SOA como padrão arquitetural de integração entre “coisas”. A abordagem utiliza a notação UML (do inglês, *Unified Modeling Language*) para representar em um nível de abstração mais alto os passos iniciais de desenvolvimento de uma aplicação IoT com base em conceitos de SoS. O uso da UML¹ nas fases iniciais permite que a modelagem tradicional seja adaptável para o desenvolvimento de aplicações IoT, permitindo que desenvolvedores já familiarizados com a UML assimilem facilmente a abordagem.

Como prova de conceito, foi desenvolvida uma simulação, resultando em uma aplicação IoT denominada “SoS Sai Dengue”. Essa aplicação foi desenvolvida seguindo os passos definidos de acordo com a proposta deste projeto de mestrado, como forma de verificar a viabilidade de desenvolvimento de acordo com as características do domínio IoT.

Para a condução deste projeto de mestrado, um mapeamento sistemático foi realizado para verificar a existência de métodos/processos/abordagens que auxiliem no processo de desenvolvimento de aplicações IoT. Por meio do mapeamento, foi possível identificar conceitos, aplicações e desafios que profissionais enfrentam durante a construção de aplicações no domínio IoT, além de verificar o estado da arte do assunto.

Observados o estado da arte e os desafios, verificou-se a necessidade de propor uma solução que auxiliasse profissionais a desenvolver aplicações IoT de acordo com as características relacionadas ao assunto. Após essa verificação, iniciou-se o estudo que compara soluções existentes e suas características para propor um guia de desenvolvimento de aplicações IoT

¹ <https://www.omg.org/spec/UML/2.5/About-UML/>

utilizando conceitos de SoS.

Por fim, deu-se início à elaboração de uma simulação como prova de conceito para verificar a aplicabilidade da abordagem desenvolvida. A simulação seguiu todos os passos descritos pela abordagem e foi documentada utilizando UML, fornecendo indícios que aplicações IoT podem ser desenvolvidas para se tornarem SoS.

1.4 Organização do trabalho

O restante da dissertação está organizada da seguinte forma: no Capítulo 2 apresenta-se a fundamentação teórica necessária sobre os assuntos desta pesquisa, sendo eles: IoT, SoS e SOA. Além desses assuntos, importantes para o entendimento desse trabalho, trabalhos relacionados são apresentados.

No Capítulo 3 é apresentado o mapeamento sistemático, descrevendo todo o processo para a obtenção e análise dos estudos importantes para o embasamento e desenvolvimento deste projeto de mestrado.

No Capítulo 4 é apresentada a abordagem proposta. Nesse capítulo são descritos os passos para a elaboração de aplicações IoT com base em SoS de acordo com os estudos realizados sobre as características de cada assunto.

No Capítulo 5 é descrita a prova de conceito da abordagem proposta, que consiste no desenvolvimento de um exemplo de aplicação denominada “SoS Sai Dengue”. O exemplo desenvolvido contém aplicações do domínio IoT e de outros domínios, e a integração entre as aplicações foi realizada por meio da utilização de *web services*.

Por fim, no Capítulo 6, são apresentadas as considerações finais da pesquisa, os desafios e dificuldades encontrados e também os trabalhos futuros.

FUNDAMENTAÇÃO TEÓRICA

2.1 Considerações Iniciais

Neste capítulo são apresentados os principais assuntos para facilitar o entendimento desta pesquisa, sendo eles: IoT, SoS e SOA. Além desses assuntos, os principais trabalhos identificados na literatura por meio do MS descrito no Capítulo 3 são abordados neste capítulo, associando as propostas dos autores com a pesquisa deste projeto de mestrado.

Sendo assim, este capítulo está organizado da seguinte forma: na Seção 2.2 são apresentados os principais conceitos, definições, características e exemplos de IoT. Na Seção 2.3 apresentam-se definições, características e exemplos de SoS. Na Seção 2.4 SOA é apresentada, descrevendo características e formas de disponibilizar serviços por meio do uso da Internet. Na Seção 2.5 são apresentados os principais trabalhos que fazem correlação com a proposta elaborada pelo proponente. Por fim, na Seção 2.6, apresentam-se as considerações finais do capítulo.

2.2 Internet das Coisas

A Internet pode ser considerada como uma entidade viva, que sempre está mudando e evoluindo. As redes, com fio e sem fio, estão por toda parte. Por meio delas, as pessoas podem se conectar e trocar diversas informações. A Internet está presente nos diversos domínios da sociedade, e tem contribuído para a evolução das diversas áreas do conhecimento (COETZEE; EKSTEEN, 2011).

Com a evolução da Internet e o crescente avanço da tecnologia, conceitos relacionados a “Internet do Futuro” têm sido pesquisados e aplicados (COETZEE; EKSTEEN, 2011). A IoT é um tópico que tem chamado bastante atenção dos atuais pesquisadores e pode ser considerada uma revolução tecnológica que pode influenciar o futuro da computação e das comunicações

(TAN; WANG, 2010).

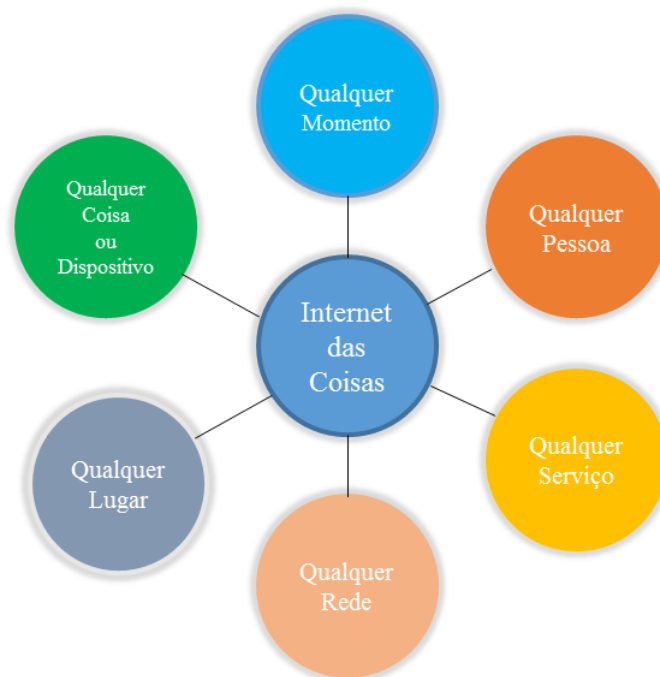
2.2.1 Definição

O conceito de “coisas” é amplo e inclui diversos dispositivos físicos, tais como telefones inteligentes (do inglês, *smart phones*), câmeras digitais, *tablets*, geladeiras, cafeteira, e sistemas de segurança automotivo. Cada um desses dispositivos físicos têm uma identificação única e podem ser conectados à Internet (DERHAMY *et al.*, 2015).

Segundo Atzori, Iera e Morabito (2010), IoT é considerada como a presença de uma variedade de “coisas” (do inglês, *things*) ou “objetos” que podem ser conectados e possuem endereçamento único, sendo capazes de interagir uns com os outros e que podem alcançar objetivos comuns.

Outras definições para IoT também são encontradas na literatura (BARI; MANI; BERKOVICH, 2013; MULANI; PINGLE, 2016). Um exemplo é a definição proposta por Evans (2011), do Grupo de Soluções para Negócios de Internet da Cisco (*Cisco Internet Business Solutions Group - IBSG*), no qual afirma que “a Internet das Coisas é o momento exato em que foram conectados à Internet mais “coisas ou objetos” do que pessoas”. Um outro exemplo é a definição proposta por Mulani e Pingle (2016), como sendo “a Internet das Coisas permite pessoas e coisas estarem conectadas a qualquer momento, em todo lugar, com tudo e todos, idealmente usando qualquer rede e qualquer serviço”. A Figura 1 ilustra essa definição. Porém, a definição adotada por este trabalho é a definição proposta por Atzori, Iera e Morabito (2010).

Figura 1 – Definição de IoT



Fonte: Mulani e Pingle (2016)

2.2.2 *Breve Histórico*

O termo “Internet das Coisas” foi inicialmente citado em 1999 por Kevin Ashton, da *Auto-ID Center*, um centro de pesquisas da *Massachusetts Institute Technology - MIT* (SURESH *et al.*, 2014). Ele atuou como pesquisador e diretor executivo entre os anos de 1999 e 2003, trabalhando juntamente com um grupo de pesquisadores no campo de Identificação por Rádio Frequência (do inglês, *Radio-Frequency Identification - RFID*) em rede e tecnologias de sensores emergentes (KRANENBURG *et al.*, 2011).

RFID é uma tecnologia de identificação que utiliza radio frequência para capturar dados, identificar objetos e controlar um determinado alvo. Um sistema RFID consiste em etiquetas (do inglês, *tags*) e leitores (do inglês, *readers*). A etiqueta é um microchip conectado com uma antena ligado ao objeto, que permite sua identificação. O leitor RFID comunica-se com a etiqueta por meio de ondas de rádio (JIA *et al.*, 2012). Além de permitir a identificação e comunicação, RFID permite o acompanhamento em tempo real dos objetos, proporcionando informações de estado e localização (SURESH *et al.*, 2014).

Porém, desde o surgimento da Internet, em 1989, o conceito de conectar “coisas” têm sido explorado por diversos pesquisadores. Um exemplo foi a torradeira, criada em 1990 por John Romkey, considerada o primeiro aparelho como um “dispositivo” ligado a rede. Ela podia ser ligada e desligada por meio do uso da Internet (SURESH *et al.*, 2014).

2.2.3 *Características da Internet das Coisas*

Na literatura, pode-se encontrar várias características para o termo IoT. Assim sendo, a Iniciativa Internet das Coisas da IEEE (*IEEE Internet of Things Initiative*) elencou um conjunto de características para facilitar a compreensão do conceito de IoT (ROTONDI, 2015). A seguir, apresentam-se as características elencadas para esse termo:

- **Interconexão de Coisas:** característica que representa a capacidade de interligação entre as “coisas”;
- **Conexão de Coisas para Internet:** essa característica sugere que as “coisas” estejam conectadas à Internet;
- **Coisas Unicamente Identificadas:** característica na qual as “coisas” são identificadas de forma única na rede;
- **Ubiquidade:** considerada a característica mais importante da IoT, a ubiquidade indica que a rede deve estar disponível em qualquer lugar e a qualquer momento;
- **Capacidade de Sensoriamento/Atuação:** essa característica permite o envolvimento de sensores e atuadores na IoT. Os sensores e atuadores são conectados nas “coisas”, permitindo a realização da detecção e atuação, gerando as funcionalidades das “coisas”;

- **Inteligência Embarcada:** essa característica sugere que os objetos sejam capazes de incorporar funções e conhecimento para executar determinada tarefa;
- **Capacidade de Comunicação Interoperável:** essa característica sugere a capacidade de comunicação de sistemas IoT baseada em protocolos de comunicação padrão e interoperáveis, devido a heterogeneidade de dispositivos;
- **Auto configuração:** essa é uma característica importante para um sistema IoT. Devido a heterogeneidade de dispositivos, tais como sensores, atuadores, dispositivos de armazenamento, dispositivo de monitoramento, telefones móveis, elementos de rede e computadores, e o elevado número de dispositivos que estão sendo conectados à Internet por meio da IoT, o controle se torna uma tarefa complicada em razão da escalabilidade limitada. Desta forma, a auto-configuração torna-se essencial para os dispositivos IoT, tanto na configuração de software ou hardware quanto na utilização de recursos como energia, largura de banda, dentre outros; e
- **Programabilidade:** essa característica permite que um dispositivo seja programável, podendo assumir comportamentos variados por um comando do usuário sem a necessidade de modificações físicas. Por exemplo, um sintetizador programável pode soar como uma série de instrumentos diferentes, dependendo da preferência do jogador, enquanto um piano tradicional pode soar somente da forma como foi fisicamente projetado para soar.

2.2.4 Aplicações

A IoT oferece um grande potencial para o desenvolvimento de aplicações em diversos domínios, tais como: transporte, ambientes inteligentes e saúde (ATZORI; IERA; MORABITO, 2010; BANDYOPADHYAY; SEN, 2011). Em geral, essas aplicações visam trazer benefícios para as pessoas do presente e do futuro, com o intuito de melhorar a qualidade de vida. Alguns exemplos de aplicações da IoT são apresentadas nas subseções seguintes (BANDYOPADHYAY; SEN, 2011; MIRKO, 2012).

2.2.4.1 Automação Doméstica

Também conhecida como *Smart Home*, a IoT pode auxiliar no controle de uma residência, fornecendo apoio para os seguintes itens: energia, aquecimento e refrigeração interna, lavadora de pratos, e lavadora de roupas, para citar alguns. Em uma *Smart Home*, o proprietário pode configurar um perfil para refrigerar ou aquecer a casa de acordo com a variação de temperatura desejada. Ele também pode determinar o desligamento da luz de um ambiente caso não haja a presença de alguém por determinado tempo. Ele ainda pode ter a opção de delegar ações de acordo com regras definidas, podendo controlar tudo isso por meio de um controle central (MIRKO, 2012).

2.2.4.2 *Indústria Médica e de Cuidados com a Saúde*

A área de saúde/medicina, se comparada com as demais áreas, obtém diversos benefícios quando utiliza recursos fornecidos pela IoT. As aplicações geradas nessa área podem auxiliar de forma significativa os médicos e pacientes. A integração de sensores com os *smartphones*, por exemplo, pode possibilitar o monitoramento de pessoas com o mal de Alzheimer, permitindo médicos analisarem a evolução da doença em tempo real (PORTAL DA MEDICINA, 2016). Isso também possibilita a informação de localização do usuário, possibilitando que a família faça o monitoramento remoto, trazendo qualidade de vida tanto para a família quanto para o portador do Alzheimer.

Outras aplicações já são realidades na área da saúde. Com o uso dos dispositivos *wearables* (dispositivos “vestíveis”), é possível realizar o monitoramento de pacientes crônicos, incluindo a verificação de sinais vitais, tais como eletrocardiograma, açúcar no sangue, oximetria, pressão, entre outros, melhorando a qualidade da saúde, ajudando no tratamento e reduzindo o volume de internações em razão de doenças como diabetes, obesidade, hipertensão, asma, e *parkinson, alzheimer* (AVI ZINS, 2015).

2.2.4.3 *Monitoramento de Ambientes*

Dispositivos identificáveis sem fio e outras tecnologias da IoT em aplicações verdes permitirão o monitoramento de áreas naturais para preservação ambiental. Além disso, também já existem aplicações de monitoramento em outros tipos de ambientes, tais como escritórios e casas, que permitem maior conforto e segurança para as pessoas. Na área urbana, por exemplo, o uso de sensores permite a transmissão de dados sobre emissão de gases poluentes, tanto de veículos quanto de indústrias. A partir desses dados, os responsáveis podem adotar medidas para diminuir os efeitos dos poluentes, protegendo o meio ambiente e contribuindo para a qualidade de vida urbana (MIRKO, 2012).

2.2.4.4 *Transporte e Trânsito*

Na indústria de transporte, a IoT pode contribuir de diferentes formas, por exemplo, nos sistemas de cobranças de tarifas de pedágio, controle de passageiros, bagagens e mercadorias em transportes comerciais, entre outros. Essas aplicações visam atender a demanda crescente por segurança. Com o uso da IoT, os engarrafamentos podem ser monitorados por meio de *smartphones*, possibilitando que o usuário escolha outra rota para continuar o trajeto com mais conforto e segurança. Além disso, é possível utilizar em aeroportos sistemas do domínio IoT para o gerenciamento de bagagens e passageiros, permitindo o acompanhamento automatizado de ambos, o que possibilita um conforto e segurança no transporte de objetos pessoais (MIRKO, 2012).

No trânsito, a utilização do GPS (*Global Position System* - Sistema de Posicionamento Global) se tornou comum no cotidiano das pessoas, facilitando a localização de determinados

lugares. Com o uso da IoT, novas informações ao sistema de navegação poderão ser adicionadas, como informações em tempo real das condições climáticas do percurso, informações de outras pessoas que estão no trânsito, e interação de com sensores em estacionamentos, informando vagas em tempo real.

Um exemplo é aplicativo Waze¹. Ele é considerado como um dos maiores aplicativos de navegação no trânsito no mundo. O Waze possui um sistema que permite a interação de usuários, compartilhando em tempo real a situação do trânsito por outros que utilizam o aplicativo. Por exemplo, uma colisão entre veículos, engarrafamento, preços de combustíveis, e informações podem ser atualizadas em tempo real. Essas informações possibilitam que os usuários economizem tempo e combustível em suas rotas. A Figura 2 ilustra as interfaces do aplicativo Waze.

Figura 2 – Interfaces do aplicativo Waze



Fonte: Elaborada pelo autor.

2.2.4.5 Reciclagem

A IoT e tecnologias sem fio podem ser usadas para ajudar em programas de apoio a coleta e reciclagem de materiais nas cidades. O gerenciamento inteligente de lixo urbano, por exemplo, pode fornecer informações úteis para a população, incentivando a coleta ambiental de forma correta. Um exemplo são as latas de lixo inteligentes que informam quando estão cheias ou próxima de estarem cheias, não sobrecarregando casas, empresas e áreas públicas. O gerenciamento inteligente também permite que as rotas de coleta sejam otimizadas de acordo com o preenchimento de cada recipiente. Além disso, com o apoio da IoT, torna-se possível a

¹ <https://www.waze.com/pt-BR>

reutilização de recursos e peças eletrônicas, tais como celulares e computadores, reduzindo o lixo eletrônico nos centros urbanos.

Nos Estados Unidos, uma empresa denominada *Bigbelly Solar*² criou lixeiras inteligentes que captam energia solar para seu sistema de aviso para recolher o lixo quando a lata de estiver cheia, aumentando a eficiência dos serviços de coleta. O sistema de energia solar da lata de lixo também permite compactar o conteúdo dentro da lata em uma proporção de 5 para 1 (50 litros se reduzem a 10, por exemplo). Essa compactação faz com que o armazenamento seja maior dentro do recipiente e também gera economia para a cidade. A Figura 3 ilustra a lixeira inteligente.

Figura 3 – Lixeira inteligente



Fonte: Belly (2003)

2.2.4.6 Agricultura e criação de animais

Na Agricultura, a IoT pode contribuir para a rastreabilidade de animais, fornecendo informações de localização em tempo real. A ação de determinar a localização ajuda a otimizar a rota de distribuição de alimento em uma grande propriedade. Além disso, permite-se verificar a qualquer momento o número de animais em um momento de surto de doença contagiosa, observando se há algum animal morto (MIRKO, 2012).

Na produção rural, a IoT permite que plantações sejam monitoradas em diferentes aspectos, como duração e nível de luz solar, temperatura, umidade e velocidade do vento. Também é possível que se obtenha dados de plantas, por exemplo, tamanho, umidade do solo e

² <http://bigbelly.com/solutions/city>

tamanho das frutas. Em uma grande propriedade, ter esses dados sobre a plantação, permite que a tomada de decisões seja melhorada para futuras plantações (MIRKO, 2012)..

2.3 Sistema-de-Sistemas

Em diversos domínios, como saúde, transporte e segurança, a prestação de serviços por meio de um sistema formado por diferentes sistemas, em geral independentes, está em amplo crescimento. Um tratamento bem sucedido de um paciente, por exemplo, pode ser resultado da interação de diversos sistemas que são operacionalmente e gerencialmente independentes, incluindo a comunicação por telefone, o acionamento de ambulância, informação e comunicação hospitalar (NIELSEN *et al.*, 2015).

Com o avanço das tecnologias de rede e comunicação, tornou-se possível que estes sistemas independentes tivessem melhores respostas e maior confiabilidade no comportamento coletivo. Dessa forma, torna-se possível conceber o conceito de “Sistema-de-Sistemas” (NIELSEN *et al.*, 2015).

2.3.1 Definições e Características

Diversas definições têm sido propostas na literatura para SoS. Dentre elas, destacam-se:

- “Um SoS é composto de sistemas interativos, chamados de sistemas componentes, que são autônomos e podem evoluir em um ambiente dinâmico. Um SoS pode ter um gerenciamento central com seus próprios objetivos e pode usar a subordinação para forçar sistemas componentes a agir como desejado” (BOUZIAT; CAMPS; COMBETTES, 2018);
- “Um SoS é um sistema grande que oferece recursos exclusivos, formados pela integração de sistemas úteis de forma independente” (ISO/IEC-24765, 2010); e
- Por fim, a definição adotada por este trabalho. “O SoS é um conjunto de sistemas colaborativamente integrados que possuem duas propriedades adicionais: independência operacional e gerencial dos componentes” (MAIER, 1998).

A necessidade de um SoS surge, segundo Maier (1998), a partir de um conjunto de características, sendo elas:

- **Independência operacional:** qualquer sistema que é parte de um SoS é independente e é capaz de operar de forma independente e útil;
- **Independência gerencial:** apesar dos SCs manterem sua independência operacional, eles também podem ser gerenciados de forma independente. Os SCs devem ser capazes de se auto gerenciar, de forma que sua existência se mantém independentemente do SoS;

- **Distribuição:** as partes que colaboram em um SoS podem estar geograficamente distribuídas, seja em uma grande área ou até mesmo localizadas em um mesmo ambiente, desde que haja um tipo de conectividade necessária para estabelecer relações, comunicação e troca de dados;
- **Desenvolvimento evolucionário:** a existência e o desenvolvimento de um SoS são evolutivos. Os objetivos e funcionalidades podem sofrer mudanças, podendo ser adicionados ou removidos com a experiência adquirida; e
- **Comportamento emergente:** Por meio da colaboração entre os SCs em um SoS, uma sinergia pode ser alcançada, de forma que o comportamento emergente do sistema satisfaz uma finalidade que não pode ser conseguida por apenas um dos SCs individuais.

Para entender e caracterizar a diferença entre SoS e sistemas monolíticos complexos, a literatura científica estabeleceu conjunto de características exclusivas para SoS, sendo elas (NIELSEN *et al.*, 2015):

- **Autonomia:** cada sistema é livre, independente e tem próprio propósito de operação;
- **Pertencente, do inglês, *belonging*:** sistemas funcionam de forma colaborativa para atingir um propósito comum maior;
- **Conectividade:** o sinergismo é permitido por meio de uma rede distribuída altamente dinâmica;
- **Diversidade:** os SCs são sistemas heterogêneos e auto suficientes que estão abertos para o aprimoramento por evolução e adaptação; e
- **Emergência:** as ações cumulativas e as interações entre os SCs de um SoS dão origem a funcionalidades que não são encontradas de forma separada nos SCs.

De acordo com Lane e Epstein (2013), as primeiras definições de SoS encontradas na literatura foram entre os anos de 1960 e 1970. Embora o termo “Sistema-de-Sistemas” não fosse normalmente usado, SoS eram desenvolvidos e implantados. Exemplos desses SoS incluem (LANE; EPSTEIN, 2013):

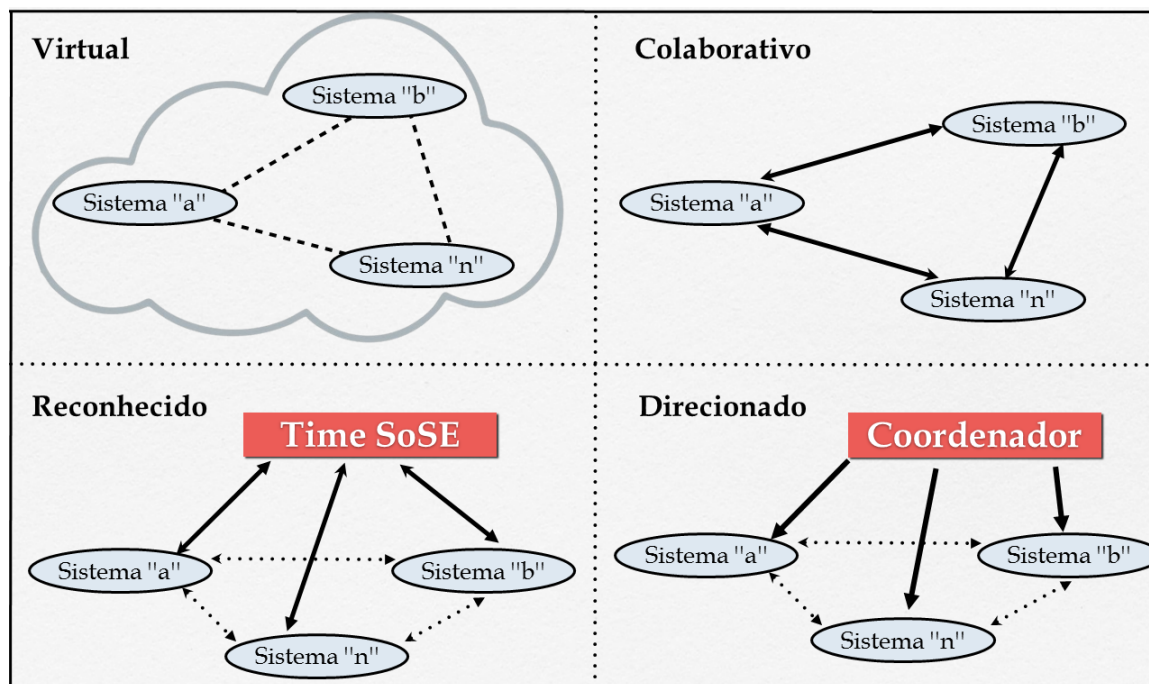
- Sistema Anti-Submarino de Guerra (do inglês, *Anti-Submarine Warfare* - ASW), usado no período da Guerra Fria, para rastrear e invadir submarinos russos;
- Sistema de Posicionamento Global (do inglês, *Global Position System* - GPS), considerado como um SoS ou um SC de um SoS; e
- Centros de comando militares.

Com esse tipo de sistema cada vez mais comum, pesquisadores e engenheiros de sistemas começaram a estudá-lo de forma a verificar abordagens estratégicas e econômicas para aumentar a capacidade de um sistema ou de um conjunto de sistemas (LANE; EPSTEIN, 2013).

2.3.2 Tipos de SoS

Diversas discussões sobre a classificação dos tipos de SoS têm sido propostas na literatura. Os tipos de SoS são fundamentos na forma em que o SoS é gerenciado e também com o grau de autonomia dos SCs (MAIER, 1998; LANE; EPSTEIN, 2013; DAHMANN, 2015). Essa classificação é fundamentada na responsabilidade e autoridade de forma a monitorar a evolução do SoS. Os tipos de SoS (Figura 4) são (LANE; EPSTEIN, 2013):

Figura 4 – Tipos de SoS



Fonte: Lane e Epstein (2013)

- **Direcionado:** um SoS direcionado é construído para atingir propósitos específicos e os SCs são gerenciados centralmente, por meio de uma equipe ou de um coordenador. O SC mantém sua capacidade de operar de forma independente, mas seu modo de operação dentro do SoS é subordinado a um coordenador central. Um exemplo desse tipo é um SoS para assistência médica, no qual uma equipe central coordena várias unidades com funcionalidades predefinidas;
- **Reconhecido:** um SoS desse tipo possui objetivos reconhecidos, um gerente ou uma equipe de gestão e recursos designados para o SoS. Porém, o gerente ou a equipe não tem

autoridade total sobre os SCs. Cada SC mantém sua independência e seus objetivos. Um exemplo é um SoS de controle e comando militar;

- **Colaborativo:** em SoS colaborativo, os SCs trabalham juntos de forma voluntária para cumprir um propósito central. Nesse tipo de SoS, não há equipe para guiar ou gerenciar as funcionalidades, podendo um SC negar ou não o fornecimento de um serviço. Um exemplo é o sistema financeiro global;
- **Virtual:** um SoS virtual carece de autoridade administrativa central e de gerenciamento sobre os SCs. Comportamento em larga escala pode surgir e pode ser desejável, porém, este tipo de SoS deve contar com o apoio de mecanismos relativamente invisíveis para mantê-lo. Um exemplo de SoS virtual é o sistema de economia nacional.

2.3.3 Aplicações

SoS possuem uma ampla gama de domínios de aplicação (NIELSEN *et al.*, 2015). Uma vez compreendido o conceito de SoS, é possível perceber que eles estão presentes em quase todos os lugares (LANE; EPSTEIN, 2013). A seguir, são apresentadas algumas aplicações de SoS (LANE; EPSTEIN, 2013; NIELSEN *et al.*, 2015):

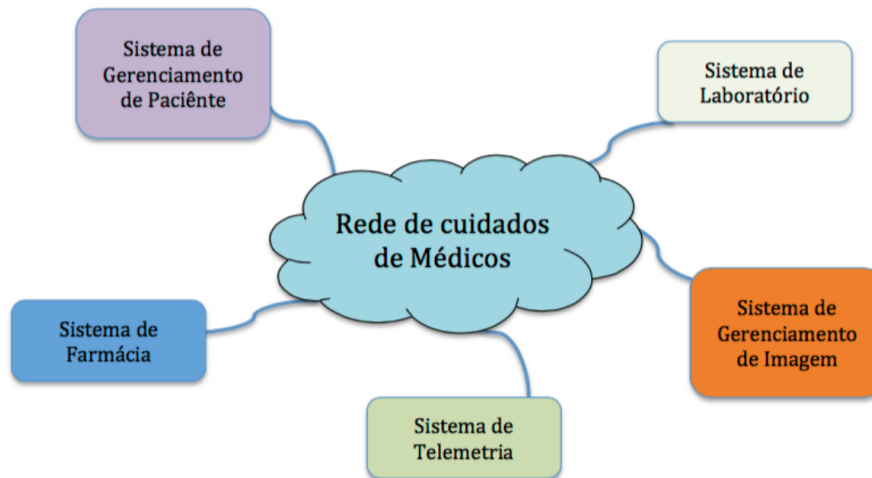
SoS em residências: sistemas de segurança doméstica ligados a empresas de segurança e também aos *smartphones*. Além disso, com as casas inteligentes (do inglês, *smart homes*), é possível fazer a integração de sistemas de segurança, sistemas climatizadores, sistemas elétricos, sistemas de alarme de incêndio, entre outros sistemas para responder a situações problemáticas.

SoS em empresas: grandes e pequenos negócios podem conter SoS. Isso pode ocorrer com a integração de vários sistemas *back office*, como sistemas de funcionários, e sistemas de folhas de pagamento. É possível, também, ter um conjunto de sistemas integrados voltados para o cliente, tais como sistemas de monitoramento de serviços, sistemas de gerenciamento de inventário e ajuda ao cliente, e sistemas de entrada de pedidos. Esse tipo de SoS é relativamente estático, pois são sistemas tipicamente conectados que interoperam uns com os outros de forma a apoiar as funções de negócios da organização. Um exemplo de SoS voltado para o cliente é um SoS de assistência médica, como mostra a Figura 5.

SoS de exploração do espaço: sistemas produzidos pelo Laboratório de Ciência de Marte (MSL - *Mars Science Laboratory*) podem ser vistos como SoS pela variedade de sistemas de pesquisas sobre plataformas de exploração que interoperam uns com os outros. Um exemplo é a câmera *Rover Mastcam*³, ilustrada na Figura 6, que interopera com outros dispositivos para registrar condições de amostras recolhidas em Marte.

³ <http://mars.jpl.nasa.gov/msl/mission/rover/>

Figura 5 – SoS de assistência médica



Fonte: Lane e Epstein (2013)

Figura 6 – Rover MastCam



Fonte: Nasa (2011)

2.4 Arquitetura Orientada a Serviços

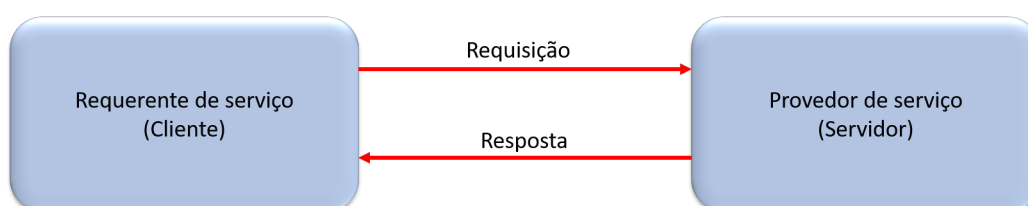
SOA é um paradigma cujo intuito é organizar e utilizar recursos distribuídos que podem ou não estar sob o controle de outros domínios (LASKEY *et al.*, 2009). As funcionalidades do software desse paradigma são empacotados em módulos autônomos e bem definidos, definidos como serviços (OLIVEIRA *et al.*, 2010).

Serviços são módulos independentes de um sistema computacional que são disponibilizados para outros sistemas, funcionando de forma independente de estado e de outros serviços, ocultando detalhes de implementação, possuindo uma interface bem definida e expondo apenas

informações relativas à sua interface. Serviços podem encapsular um método simples ou um grande processo envolvendo múltiplos colaboradores, utilizam protocolo padrão e aberto para comunicação, como o HTTP (do inglês *Hyper Text Transfer Protocol*) (LASKEY *et al.*, 2009; SOUZA *et al.*, 2006).

De forma simplificada, como ilustrado na Figura 7, a arquitetura SOA pode ser vista como uma aplicação cliente/servidor de uma nova maneira, no qual aquele que faz a requisição de um serviço (cliente) envia uma mensagem para quem está disponibilizando o serviço (servidor) e o provedor de serviço envia uma resposta, podendo ser uma informação que foi buscada ou confirmando alguma ação executada (SOUZA *et al.*, 2006).

Figura 7 – Visão simplificada de SOA



Fonte: Souza *et al.* (2006)

2.4.1 Características

A utilização de SOA traz diversas vantagens no desenvolvimento de software, destacando-se entre elas (SOUZA *et al.*, 2006; FURTADO *et al.*, 2009):

- **Interoperabilidade:** serviços devem ser elaborados utilizando-se tecnologias padronizadas e disponíveis a várias plataformas de software;
- **Fraco acoplamento:** um dos princípios fundamentais de SOA, o baixo acoplamento significa projetar serviços minimizando independências, de forma que modificações ou falhas em um serviço possam causar baixos impactos;
- **Granularidade adequada:** o encapsulamento de funcionalidades no nível de serviço permite uma granularidade nos componentes básicos da arquitetura;
- **Composição:** serviços podem ser compostos para formar outros serviços com um nível maior de abstração;
- **Reusabilidade:** por possuir fraco acoplamento e serem módulos independentes de código, os serviços podem ser facilmente reutilizados, aumentando a produtividade no desenvolvimento de software; e
- **Ubiquidade:** os serviços devem estar disponíveis e serem acessados em qualquer lugar e a qualquer momento.

2.4.2 Web Services

Considerada uma tecnologia relativamente nova na computação, os *web services* receberam ampla aceitação na implementação de SOA. Isso se dá ao fato de que os *web services* oferecem uma abordagem de computação distribuída para que aplicações heterogêneas pudessem interagir por meio do uso da Internet. *Web services* são independentes de linguagem de programação, sistema operacional e hardware, promovendo assim baixo acoplamento entre provedor e consumidor de serviços (SOUZA *et al.*, 2006).

De acordo com o *Web Services Architecture Working Group*⁴, o qual é parte do *W3C Web Services Activity*, a definição de *web service* é: “um sistema de software projetado para apoiar a interação máquina a máquina de forma interoperável. Ele possui uma interface descrita em formato processável por máquina (especificamente WSDL). Outros sistemas interagem com o *web service* de maneira prescrita por sua descrição utilizando mensagens SOAP, normalmente transmitidas usando HTTP com uma serialização XML em conjunto com outros padrões relacionados à web”.

De acordo com a definição da W3C e também de Furtado *et al.* (2009), os elementos padrão dos *web services* são SOAP, UDDI e WSDL, sendo definidos como:

- **WSDL** (*Web Services Description Language* - Linguagem de Descrição de Serviços Web) é uma linguagem baseada em XML utilizada para descrever os serviços web, ou seja, um documento contendo a descrição do serviço, como acessá-lo e quais operações ou métodos estão disponíveis;
- **SOAP** (*Simple Object Access Protocol* - Protocolo Simples de Acesso a Objeto) é o protocolo de comunicação que permite a troca de mensagens de *web services*. É descrito em formato XML, sendo independente de plataforma e de implementação. SOAP permite baixo acoplamento entre o provedor e consumidor de serviço, além de proporcionar a comunicação entre serviços de diferentes organizações; e
- **UDDI** (*Universal Description, Discovery and Integration* - Descrição, Descoberta e Integração Universal) é um protocolo baseado em XML que oferece um diretório distribuído com listas de negócios na Internet e descoberta de serviços.

Além do SOAP, outro protocolo de comunicação utilizado para comunicação entre *web services* é o REST (do inglês, *Representation State Transfer* - Transferência de Estado Representacional). Esse protocolo de troca de mensagens é mais recente em relação ao SOAP e surgiu com o objetivo de simplificar o acesso aos *web services*. REST é baseado no protocolo HTTP e permite outros formatos para a representação de dados além do XML, como o JSON e texto puro (PAUTASSO; ZIMMERMANN; LEYMAN, 2008).

⁴ <https://www.w3.org/2002/ws/arch/>

Os recursos de um *web service* do tipo REST são identificados por meio da utilização de URIs (do inglês, *Uniform Resource Identifier* - Identificador Único do Recurso) e manipulados com a utilização dos métodos HTTP, sendo possível criar, atualizar, excluir e realizar outras operações (WAGH; THOOL, 2012). A Tabela 1 apresenta os métodos do protocolo HTTP.

Tabela 1 – Métodos HTTP

Método HTTP	Descrição
GET	Obter dados de um recurso
POST	Criar um novo recurso
PUT	Substituir os dados de um determinado recurso
PATCH	Atualizar parcialmente um determinado recurso
DELETE	Excluir um determinado recurso
HEAD	Similar ao GET, porém utilizado para obter os cabeçalhos de resposta, sem os dados em si
OPTIONS	Obter quais manipulações podem ser realizadas em um determinado recurso

Fonte: Elaborada pelo autor.

2.4.2.1 Estrutura do web service

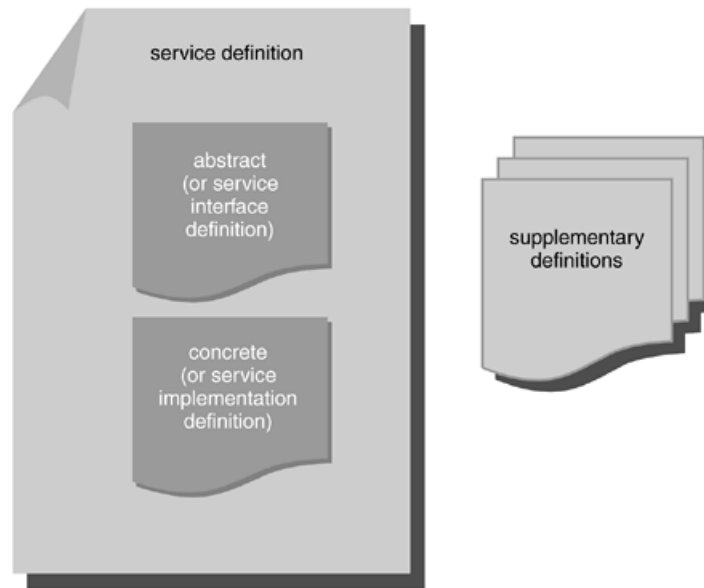
Um *web service* é descrito por meio de documentos de descrição, servindo como blocos de elaboração para uma descrição de serviço (ERL, 2004). Na Figura 8 são apresentados documentos que contêm as definições dos serviços.

Como ilustrados na Figura 8, *abstract* contém a descrição da *interface* de um *web service*, independentemente dos detalhes da implementação. Já *concret* contém a localização e informações específicas de implementação sobre um *web service*. Constituem como parte do *concret* os elementos de ligação (*binding*), serviço (*service*) e ponto de término (*endpoint*). Tanto o conteúdo do *abstract* quanto o conteúdo do *concret* são descritos no documento WSDL. Além dessas definições, informações adicionais podem ser incluídas por meio de outros documentos, fornecendo informações complementares (*supplementary definitions*) (ERL, 2004).

2.4.2.2 Papéis do web service

Serviços podem assumir papéis diferentes de acordo com a diferenciação de cenários no qual existe a interação. Isso pode ser definido de acordo como o serviço é visualizado e também de como a tarefa está em execução em determinado momento, podendo ter uma das funções a seguir (ERL, 2004):

Figura 8 – Conteúdo da descrição de um serviço



Fonte: Erl (2004)

- **Provedor de serviço:** um *web service* expõe uma interface pública que pode ser chamada por quem deseja consumir o serviço;
- **Solicitante de serviço:** um consumidor de serviço solicitando um *web service* específico;
- **Intermediário:** situação em que o *web service* recebe a mensagem de um consumidor de serviço e passa para outro provedor de serviços;
- **Remetente inicial:** considerado como um consumidor de serviço, este termo é utilizado para diferenciar o primeiro *web service* que envia uma mensagem; e
- **Receptor final:** representa o último *web service* a receber uma mensagem de solicitação de serviço.

2.5 Trabalhos Relacionados

Nesta seção são discutidos os estudos que trazem soluções que auxiliam o desenvolvimento de aplicações IoT, identificando características importantes para o domínio e os desafios enfrentados por profissionais durante a construção de aplicações IoT. O objetivo é revisar o estado da arte e relacionar o trabalho desenvolvido com pesquisas que apresentam, de alguma forma, aspectos importantes para o projeto aqui apresentado.

Patel e Cassou (2015) propuseram um estudo cujo propósito é facilitar o desenvolvimento de aplicações IoT para os *stakeholders*, inspirando-se na abordagem MDD e desenvolvendo a partir da macroprogramação de redes de sensores. Segundo os autores, o objetivo do trabalho foi

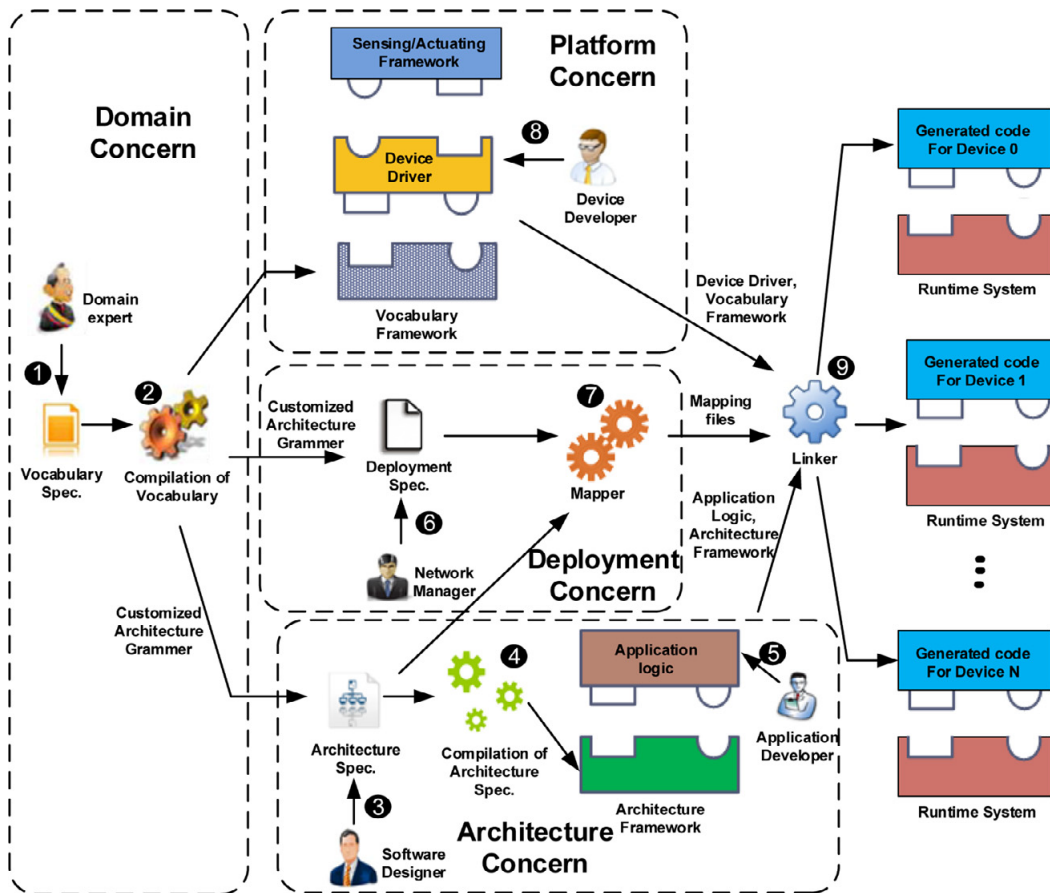
alcançado por meio da separação do processo de desenvolvimento de aplicações IoT em diferentes interesses para cada *stakeholder* e também pela integração de um conjunto de linguagens de alto nível para especificá-las. Especificamente, o estudo propôs:

1. Metodologia de desenvolvimento, que define uma sequência de etapas para o desenvolvimento de aplicações IoT, separados em quatro diferentes domínios, sendo eles: domínio, funcional, implantação e plataforma. Segundo os autores, essa separação permite que os *stakeholders* lidem com cada interesse individualmente e os reutilizem em outras aplicações; e
2. *Framework* de desenvolvimento, construído para apoiar as ações dos *stakeholders* tomadas com os passos da metodologia de desenvolvimento, oferecendo um conjunto de linguagens de modelagem e técnicas de automação em diferentes fases no desenvolvimento de aplicações IoT, incluindo:
 - Um conjunto de linguagens de modelagem, integrando três linguagens de modelagem que abstraem a complexidade com relação à escala e à heterogeneidade, sendo elas: i) *Srijan*⁵ *Vocabulary Language* (SVL) para descrever características específicas de domínio de uma aplicação IoT; ii) *Srijan Architecture Language* (SAL) para descrever funcionalidades específicas de domínio de uma aplicação IoT; e iii) *Srijan Deployment Language* (SDL) para descrever características específicas de implantação consistindo em informações sobre o ambiente físico quando dispositivos são implantados;
 - Técnicas de automação, contendo técnicas de geração de código, mapeamento de tarefas e de ligação. A geração de código oferece apoio à fase de desenvolvimento da aplicação, produzindo uma estrutura de programação que reduz o esforço de especificar os detalhes dos componentes de uma aplicação IoT. O mapeamento de tarefas e a ligação apoiam, juntos, a fase de implantação, produzindo código específico para o dispositivo.

A metodologia proposta pelos autores é apresentada na Figura 9. A metodologia separa as responsabilidades dos *stakeholders* em cinco diferentes papéis: especialista de domínio (*domain expert*), projetista de software (*software designer*), desenvolvedor de aplicações (*application developer*), desenvolvedor de dispositivo (*device developer*) e gerente de rede (*network manager*). As habilidades e responsabilidades de cada papel podem ser vistas na Tabela 2.

⁵ *Srijan* é um sânscrito da palavra “criação”

Figura 9 – Processo geral de desenvolvimento de aplicações IoT



Fonte: Patel e Cassou (2015)

Tabela 2 – Papéis dos stakeholders no desenvolvimento de aplicações IoT

Papel	Habilidades	Responsabilidades
Domain expert	Compreender os conceitos de domínio, incluindo os tipos de dados produzidos pelos sensores, consumidos pelos atuadores, acessados de locais de armazenamento, interações do usuário, e como o sistema é dividido em regiões	Especificar o vocabulário de um domínio de aplicação a ser utilizado por aplicações no domínio
Software designer	Conceitos de arquitetura de software, incluindo o uso adequado dos modos de interação, método publicação-assinatura, comando, e solicitação-resposta para o uso na aplicação	Definir a estrutura de uma aplicação IoT especificando os componentes de software e seus relacionamentos de geração, consumo e comando
Application developer	Hábil para projetar algoritmos e utilizar linguagens de programação	Desenvolver a lógica de aplicação dos serviços computacionais na aplicação
Device developer	Entendimento das entradas e saídas, e protocolos dos dispositivos individuais	Escrever drivers para os sensores, atuadores, dispositivos de armazenamentos, e aplicações de usuários finais utilizados no domínio
Network manager	Entendimento da área específica do destino onde a aplicação deve ser implantada	Instalar a aplicação no sistema. Esse processo pode envolver a geração de binários ou bytecode, e a configuração de middleware

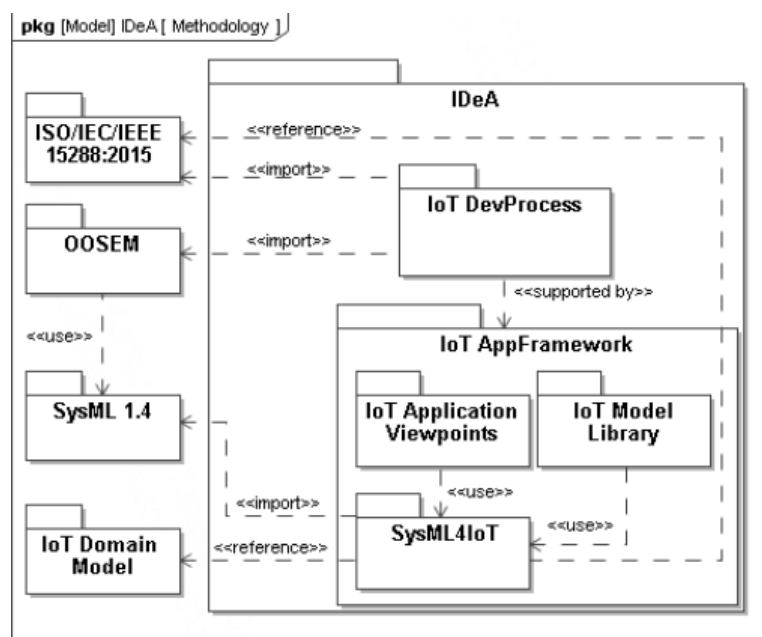
Fonte: Patel e Cassou (2015)

Costa, Pires e Delicato (2016) elaboraram uma metodologia de engenharia de sistemas baseada em modelos para o desenvolvimento de aplicações IoT, denominado *IDeA - IoT DevProcess & AppFramework*. Os objetivos dessa metodologia são:

- fornecer abstrações em alto nível por meio da meta-modelagem para auxiliar a heterogeneidade de componentes de software e dispositivos de hardware no modelo de sistema;
- prover mecanismos para o projeto de aplicações IoT multidisciplinares utilizando *views* e *viewpoints* para lidar com diferentes *stakeholders* envolvidos no processo; e
- elaborar um método a partir de padrões existentes e consolidados com atividades consideradas relevantes no projeto de aplicações IoT.

IDeA é composto de um método, denominado *IoT DevProcess* e uma ferramenta de apoio, denominada *AppFramework*. O *IoT DevProcess* é uma extensão da OOSEM (*Object-Oriented Systems Engineering*) adaptado especialmente para o projeto de aplicações IoT e concebido para ser aplicado no contexto do processo técnico da ISO/IEC/IEEE 15288, um método padrão que utiliza SysML para dar suporte as fases de análise, especificação, projeto, verificação e validação de sistemas complexos. Para dar suporte as atividades do *IoT DevProcess*, o *IoT AppFramework* provê um perfil SysML, denominado de *SysML4IoT*, que é baseado em conceitos bem definidos de IoT. A metodologia do *IDeA* é apresentada na Figura 10.

Figura 10 – Metodologia do *IDeA*

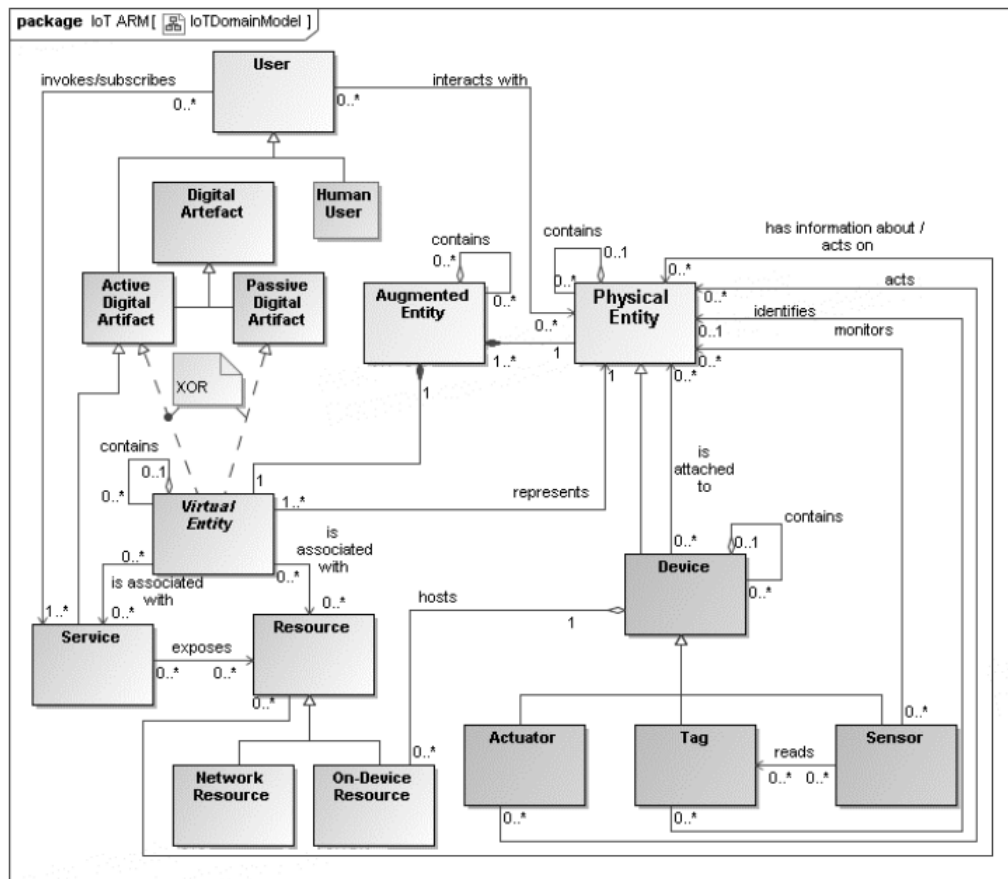


Fonte: Costa, Pires e Delicato (2016)

O *SysML4IoT* fornece abstrações para especificar diferentes tipos de dispositivos de hardware, software, serviços de software, fluxos de dados, dentre outros. Essas abstrações são

baseadas em um modelo de referência para IoT, o IoT ARM (*Internet of Things Architecture Reference Model*), elaborado por pesquisadores europeus com intuito de padronizar o domínio IoT como modelo de referência. O modelo de domínio IoT proposto por Bassi *et al.* (2013) é apresentado na Figura 11.

Figura 11 – Modelo de domínio IoT



Fonte: Costa, Pires e Delicato (2016)

De acordo com a Figura 11, a **Entidade Física** é uma parte identificável do ambiente físico que é de interesse para os *stakeholders* da aplicação (por exemplo, carro, aparelho eletrônico). **Entidades Virtuais** são representações virtuais das Entidades Físicas. Uma Entidade Física pode ser um **Dispositivo** (artefatos técnicos como Arduino ou Raspberry Pi), um **Atuador** (que altera o estado de uma entidade física, como ligar/desligar), uma **Tag** (que identifica uma Entidade Física, como por exemplo, RFID) ou um **Sensor** (que fornece informações, conhecimento ou dados sobre a Entidade Física que eles monitoram, como, por exemplo, o sensor de temperatura). A interação entre artefatos digitais e entidades físicas é mediada por um **Serviço** de software, que expõe os **Recursos** (componentes de software que fornecem dados ou são usados na atuação em Entidades Físicas). Os **Recursos no Dispositivo** são recursos que estão hospedados em Dispositivos locais, enquanto os **Recursos de Rede** são recursos que estão disponíveis em algum lugar na rede, por exemplo em servidores.

Os autores também afirmam que existem múltiplos interesses ou conhecimentos dos *stakeholders* durante o processo de desenvolvimento de aplicações IoT que podem gerar conflitos. Dessa forma, faz-se necessário a separação de interesses durante o ciclo de vida do desenvolvimento de uma aplicação IoT. A Tabela 3 descreve as diferentes caracterizações dos *stakeholders* envolvidos no desenvolvimento de aplicações IoT, bem como suas habilidades e responsabilidades.

Tabela 3 – Diferentes *Stakeholders* no desenvolvimento de aplicações IoT

Stakeholder	Habilidades	Responsabilidades
Especialista de Dispositivo	Profundo entendimento dos protocolos dos dispositivos, conceitos do Modelo de Domínio IoT, e SysML	Especificar, implementar, e publicar os serviços dos dispositivos
Especialista de Domínio	Entendimento do domínio de aplicação e conceitos do Modelo de Domínio IoT	Definir os requisitos e o escopo da aplicação
Engenheiro de Requisitos IoT	Profundo entendimento sobre engenharia de requisitos, conceitos do Modelo de Domínio IoT, e SysML	Elicitar os requisitos da aplicação e do dispositivo
Engenheiro de Aplicação IoT	Profundo entendimento sobre Engenharia de Sistemas, conceitos do Modelo de Domínio IoT, e SysML	Especificar a estrutura de uma aplicação IoT
Gerente de Implantação IoT	Profundo entendimento de plataformas e áreas onde a aplicação será implantada, conceitos do Modelo de Domínio IoT, e UML	Implantar a aplicação

Fonte: [Costa, Pires e Delicato \(2016\)](#)

2.6 Considerações Finais

Neste capítulo foram apresentados os principais conceitos estudados para a realização desta pesquisa, sendo eles IoT e SoS. O entendimento desses assuntos é fundamental para a compreensão deste trabalho, pois visam auxiliar desenvolvedores e pesquisadores a como elaborar aplicações IoT com base em conceitos de SoS, além da utilização da arquitetura SOA para a integração entre os sistemas presentes em um SoS, como afirma [Vargas, Gottardi e Braga \(2016\)](#). Além disso, dois estudos foram descritos que apresentam aspectos importantes no desenvolvimento de aplicações IoT e que podem ser relacionados a esta pesquisa.

No próximo capítulo é descrito o mapeamento sistemático, cujo foco foi a busca por estudos que apresentem soluções que apoiem o desenvolvimento de aplicações IoT. O mapeamento realizado teve como objetivo obter uma visão dos desafios enfrentados por pesquisadores no desenvolvimento desse tipo de aplicação, além de investigar a existência de casos de desenvolvimento de aplicações IoT que se assemelhem com SoS.

ABORDAGENS PARA O DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET DAS COISAS - UM MAPEAMENTO SISTEMÁTICO

3.1 Considerações Iniciais

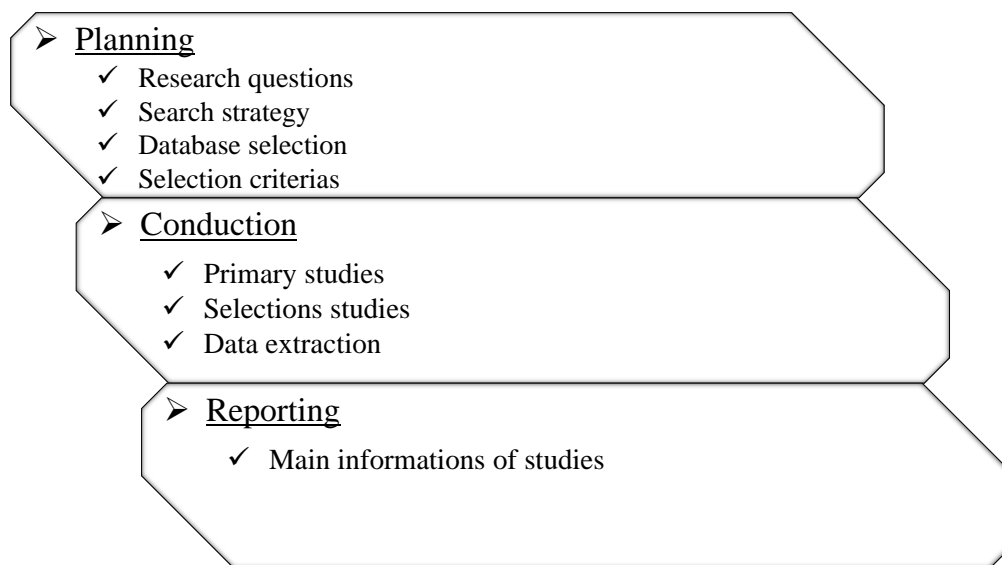
Para identificar estudos que utilizaram abordagens/métodos/processos que apoiem o desenvolvimento de aplicações IoT, realizou-se um mapeamento sistemático da literatura. Um mapeamento sistemático é um tipo de pesquisa muito utilizada quando há um cenário amplo e cujo principal objetivo é reunir o maior número de informações disponíveis sobre uma determinada área do conhecimento (KITCHENHAM; CHARTERS, 2007; PETERSEN *et al.*, 2008), nesse caso IoT.

O presente capítulo aborda o mapeamento sistemático realizado, descrevendo o processo realizado para encontrar estudos relevantes da área de pesquisa. O capítulo encontra-se estruturado da seguinte maneira: na Seção 3.2, apresentam-se as principais etapas realizadas, sendo elas: i) Planejamento, onde são definidos o objetivo e questões de pesquisa, a *string* de busca, a seleção das bases de dados, os critérios de inclusão e os critérios de exclusão; ii) Condução, nos quais foram executados a *string* de busca nas bases de dados e a seleção dos estudos encontrados; e iii) Sumarização, no qual fez-se a extração das principais informações dos estudos encontrados, que foram então analisadas para responder as questões de pesquisas definidas. Por fim, na Seção 3.3, apresentam-se as considerações finais deste capítulo.

3.2 Mapeamento Sistemático

O mapeamento foi conduzido de acordo com as diretrizes propostas por Kitchenham e Charters (2007), seguindo três diferentes fases, sendo elas: planejamento, condução e sumarização dos dados. Essas fases podem ser observadas na Figura 12.

Figura 12 – Fases para a realização do mapeamento sistemático



Fonte: Elaborada pelo autor.

- **Planejamento:** o objetivo dessa fase foi verificar a necessidade da realização do mapeamento e, conseqüentemente, definir o protocolo para sua condução. O protocolo inclui: questões de pesquisa, *string* de busca, seleção das bases de dados, critérios de inclusão e critérios de exclusão;
- **Condução:** o objetivo dessa fase foi a busca de estudos primários para análise seguindo o protocolo definido. Nessa fase também foi realizada a seleção e extração de dados necessários para responder as questões de pesquisa; e
- **Sumarização dos Dados:** o objetivo dessa fase foi responder as questões de pesquisa propostas dos estudos selecionados.

Para auxiliar a execução desse mapeamento sistemático, utilizou-se o Parsifal¹, uma ferramenta *online* que facilitou a execução de todas as fases do mapeamento, que são detalhadas a seguir.

¹ <https://parsif.al/>

3.2.1 Planejamento

Na fase de planejamento, o protocolo do mapeamento foi definido de forma a especificar o objetivo da pesquisa, as questões de pesquisa, a *string* de busca, a seleção das bases de dados, os critérios de inclusão e os critérios de exclusão. Todas essas etapas são descritas a seguir.

3.2.1.1 Objetivo e questões de pesquisa

O mapeamento sistemático teve por objetivo identificar os tipos de soluções (por exemplo, processos, métodos, abordagens, entre outros) existentes na literatura que apoiassem o desenvolvimento de aplicações para IoT, verificando quais fases do ciclo de vida no processo de desenvolvimento de software o tipo de solução identificada aborda.

Sendo assim, questões de pesquisas (do inglês *Research Question* - RQ) foram elaboradas para a realização do mapeamento sistemático. Para este estudo, definiram-se três questões de pesquisa, sendo elas:

- **RQ 1:** Quais são os tipos de soluções (por exemplo, processos, métodos, abordagens, entre outros) existentes na literatura para apoiar o desenvolvimento de aplicações para IoT?

RQ 1.1: Dentre as soluções, há alguma em que se aborda o conceito de Sistema-de-Sistemas (SoS)?

- **RQ 2:** Quais fases do ciclo de vida de desenvolvimento do software o tipo de solução identificada se aplica?

Para uma questão de pesquisa ser bem formulada, ela precisa ser composta e também analisada em diferentes pontos de vista. Dessa forma, para este mapeamento sistemático, foi definido o PICO (População, Intervenção, Comparação e Resultados) (KITCHENHAM; CHAR-TERS, 2007). População identifica o grupo a ser observado na intervenção; Intervenção refere-se ao que será observado no contexto deste mapeamento; Comparação define o que será comparado no contexto da avaliação sistemática e, por último, os Resultados são as conclusões definidas. O PICO deste mapeamento sistemático é apresentado como segue:

- **População:** Estudos primários que apresentam tipos de soluções (por exemplo, processos, métodos, abordagens, entre outros) que apoiem o desenvolvimento de aplicações IoT;
- **Intervenção:** Quais fases os tipos de soluções apoiam, quais os benefícios, quais os tipos de soluções e quais recursos da engenharia de software os tipos de soluções utilizam;
- **Comparação:** Não se aplica neste mapeamento, pois o objetivo é reunir informações de tipos de soluções que apoiem o desenvolvimento de aplicações IoT e extrair suas principais características sem realizar um estudo comparativo entre os estudos; e

- **Resultados:** Uma visão geral dos tipos de soluções propostas na literatura para o desenvolvimento de aplicações IoT, evidenciando suas características, benefícios, entre outros.

3.2.1.2 String de busca

Após a definição das questões de pesquisa, quatro palavras-chave foram identificadas para a busca de estudos primários na literatura, sendo elas: “internet of things”, “method”, “development” e “software”. Também foram considerados os sinônimos e variações de cada palavra-chave, com exceção da palavra-chave “software”. Na Tabela 4, apresentam-se as palavras-chave e seus sinônimos.

Tabela 4 – Palavras-chave e sinônimos

Palavra-chave	Sinônimo
Internet of things	“IoT”, “internet-of-things”
Method	“technique”, “approach”, “process”, “methodology”
Development	“modeling”, “modelling”, “creation”, “developing”, “construction”
Software	-

Fonte: Elaborada pelo autor.

A *string* resultante, criada de acordo com as quatro palavras-chaves definidas e seus sinônimos, é apresentada na Tabela 5. Os principais termos são conectados com o operador lógico AND e as variações são conectadas com o operador lógico OR.

Tabela 5 – String de busca

(“internet of things” OR “iot” OR “internet-of-things”) AND (“method” OR “technique” OR “approach” OR “process” OR “methodology”) AND (“development” OR “modelling” OR “modeling” OR “creation” OR “developing” OR “construction”) AND (“software”)

Fonte: Elaborada pelo autor.

3.2.1.3 Seleção das bases de dados

As bases de dados foram selecionadas considerando a experiência de pesquisadores conforme recomendado por Petersen, Vakkalanka e Kuzniarz (2015) e também em fontes eletrônicas que possuem relevância para engenheiros de software (KITCHENHAM; CHARTERS, 2007). Na Tabela 6, apresentam-se as bases de dados utilizadas e seus respectivos endereços eletrônicos.

Tabela 6 – Base de dados selecionadas

Base de dados	URL
IEEE Xplore	http://ieeexplore.ieee.org/
ACM Digital Library	http://dl.acm.org/
Scopus	http://https://www.scopus.com/
Science Direct	http://www.sciencedirect.com
Web of Science	http://webofknowledge.com/
El Compendex	http://www.engineeringvillage.com/

Fonte: Elaborada pelo autor.

3.2.1.4 Critérios de inclusão e exclusão

Os critérios de inclusão e exclusão são utilizados para avaliar se os estudos primários serão considerados na análise do mapeamento sistemático. Para selecionar os estudos primários por meio das bases de dados, foram utilizados os seguintes critérios:

Critérios de inclusão (IC - do inglês *Inclusion Criteria*):

- **IC1:** Estudos que apresentam tipos de soluções (por exemplo, processos, métodos, abordagens, entre outros), utilizando ou não abordagens da engenharia de software, para apoiar o desenvolvimento de aplicações IoT;
- **IC2:** Estudos escritos em inglês e/ou português.

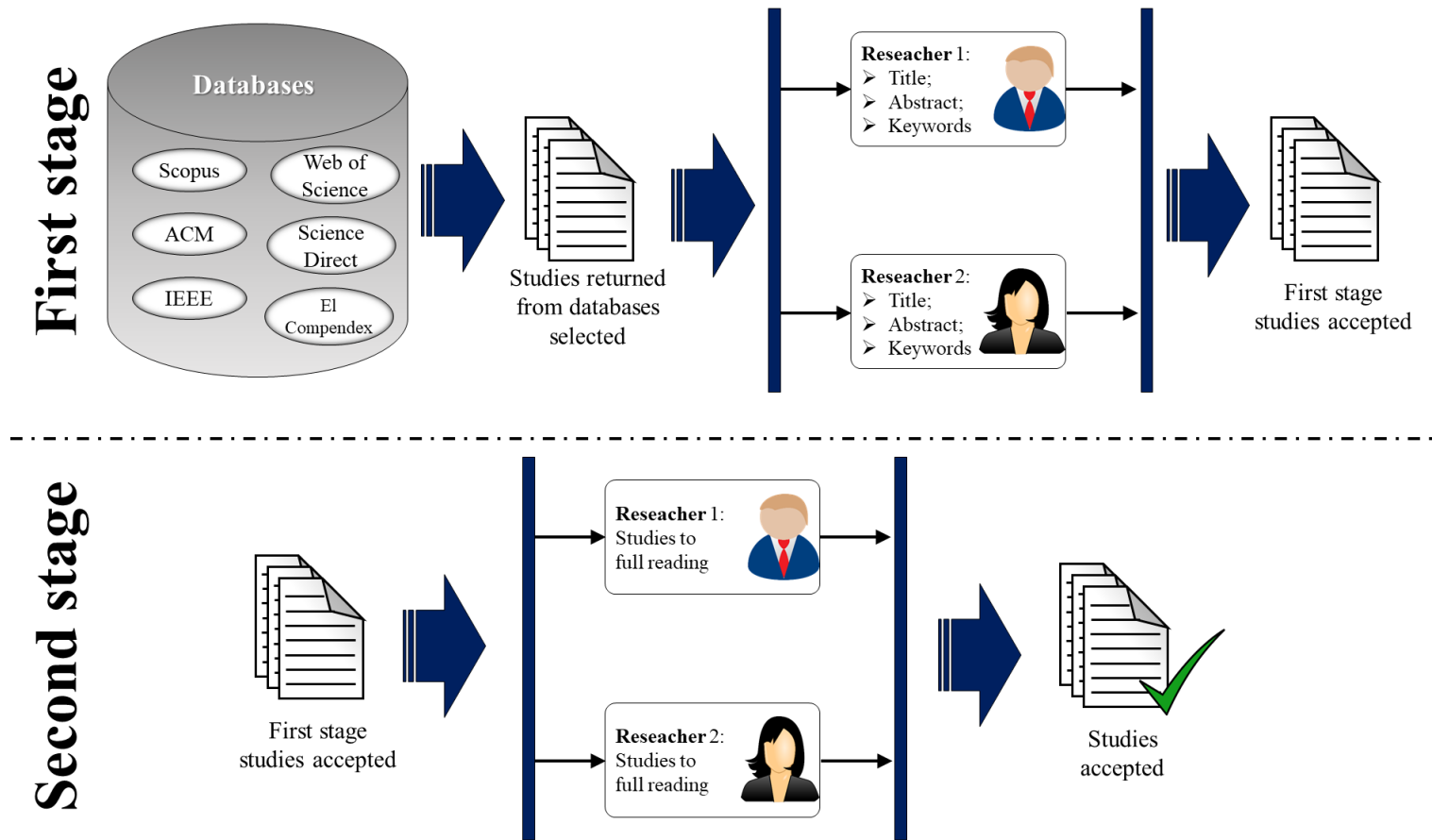
Critérios de exclusão (EC - do inglês *Exclusion Criteria*):

- **EC1:** Estudos que não apresentam tipos de soluções (por exemplo, processos, métodos, abordagens, entre outros) que apoiem o desenvolvimento de aplicações para IoT;
- **EC2:** Estudos que descrevem eventos, índice; ou programação; e
- **EC3:** Estudos que não são escritos em inglês e/ou português.

3.2.2 Condução

O processo de condução ocorreu em duas etapas: i) execução da *string* de busca nas bases de dados selecionadas e seleção dos estudos primários por meio da análise do título, *abstract* e palavras-chave; e ii) seleção de estudos encontrados na primeira etapa por meio de leitura completa. As etapas realizadas são ilustradas na Figura 13. O MS foi executado entre os meses de Junho a Setembro de 2017.

Figura 13 – Passos do Mapeamento Sistemático



Fonte: Elaborada pelo autor.

Na primeira etapa, a pesquisa foi executada em cada base de dados por meio da *string* de busca definida. A *string* de busca foi adaptada de acordo com as particularidades de cada base de dados, porém a lógica estrutural da *string* de busca, como apresentado na Tabela 5, se manteve. No total, foram obtidos 1971 estudos primários após a execução da *string* nas bases de dados, conforme apresentado na Tabela 7. Não houve uma limitação de ano na busca por estudos primários, uma vez que o assunto relacionado à IoT é considerado novo no meio acadêmico e industrial. De acordo com Aziz, Sheikh e Felemban (2016), as pesquisas sobre aplicações IoT em diferentes domínios têm originado diferentes soluções no meio acadêmico e industrial.

Tabela 7 – Quantidade de trabalhos retornados por base de dados

Base de dados	Estudos retornados
IEEE Xplore	361
ACM Digital Library	257
Scopus	637
Science Direct	30
Web of Science	187
EI Compendex	499
TOTAL	1971

Fonte: Elaborada pelo autor.

Após a execução da *string* de busca e obtenção dos estudos, deu-se início ao primeiro processo de seleção. Dos 1971 estudos primários retornados pelas bases de dados, 808 estudos duplicados foram excluídos, ou seja, estudos que foram indexados por mais de uma base de dados. Em seguida, 1120 estudos foram excluídos por meio da leitura dos títulos e *abstracts* aplicando-se os critérios de inclusão e exclusão, totalizando em 43 estudos para leitura completa.

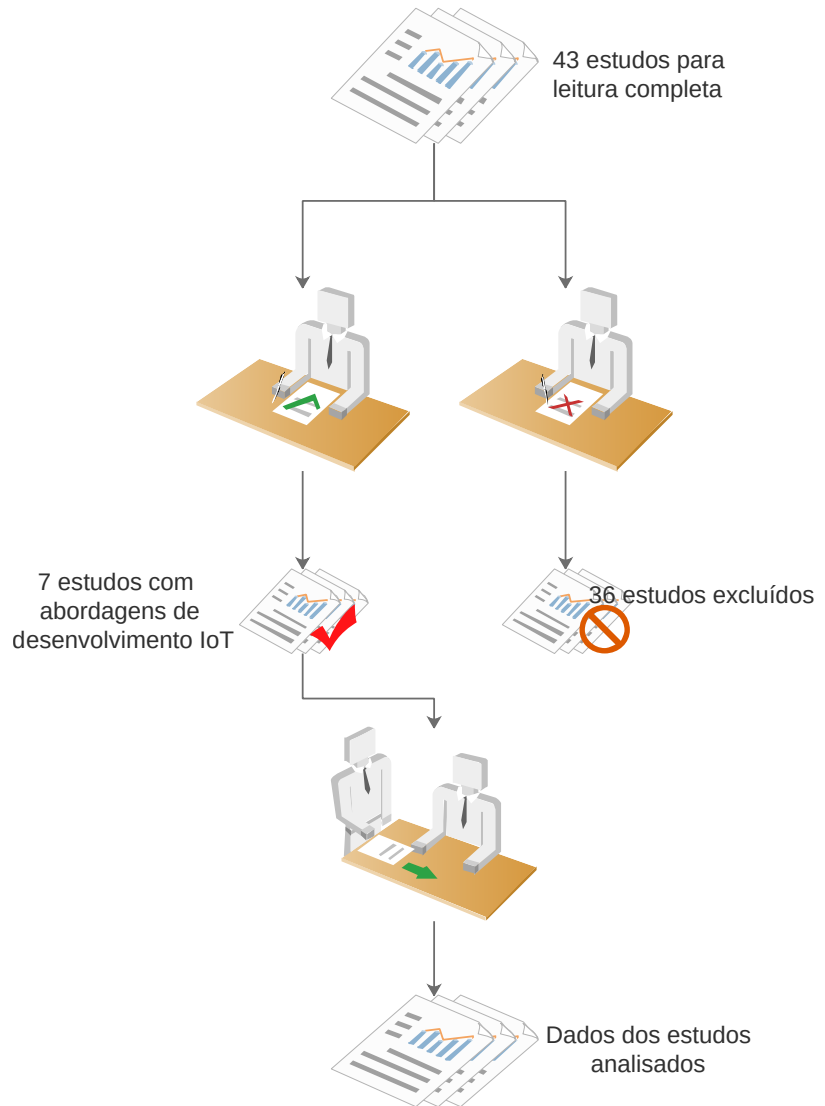
Na segunda etapa de seleção, o processo de leitura completa foi realizado para analisar e selecionar estudos relevantes. Desse processo, 36 artigos foram excluídos por meio da aplicação dos critérios de inclusão e exclusão, restando sete estudos que abordam formas de apoiar o desenvolvimento de aplicações IoT. O processo de seleção dos estudos pode ser observado na Figura 14. Além disso, na Tabela 8, apresentam-se os estudos selecionados para análise, nomeados com IDs de S1 a S7 para facilitar a referência.

3.2.3 Sumarização

A fase de sumarização teve por objetivo analisar as principais informações dos estudos que foram selecionados na segunda etapa de seleção de estudos deste mapeamento sistemático para responder as questões de pesquisa.

RQ 1: *Quais são os tipos de soluções (por exemplo, processos, métodos, abordagens, entre outros) existentes na literatura para apoiar o desenvolvimento de aplicações para IoT?*

Figura 14 – Processo de seleção dos estudos primários



Fonte: Elaborada pelo autor.

Diversas soluções têm sido propostas, como metodologias, *frameworks*, arquiteturas, entre outras. Além disso, foram encontrados estudos sobre características e abstrações que ajudam no processo de desenvolvimento de aplicações e comparações entre os métodos de desenvolvimento existentes, para que possam ser adaptados ao paradigma de desenvolvimento da IoT.

Tabela 8 – Estudos selecionados para análise e extração de dados

ID	Título	Publicação
S1	Enabling high-level application development for the internet of things	2015
S2	Low cost IoT software development - Ingredient transformation and interconnection	2015
S3	Model driven development for internet of things application prototyping	2013
S4	Modeling IoT Applications with SysML4IoT	2016
S5	Parametric Statecharts: Designing Flexible IoT Apps: Deploying Android M-health Apps in Dynamic Smart-homes	2017
S6	Towards a CVL process to develop agents for the IoT	2014
S7	Towards a Development Methodology for Smart Object-Oriented IoT Systems: a Metamodel Approach	2015

Fonte: Elaborada pelo autor.

O estudo S1 (PATEL; CASSOU, 2015) traz uma metodologia que separa o desenvolvimento de aplicações IoT em diferentes interesses, de acordo com as habilidades dos *stakeholders*. O estudo também apresenta a implementação de um *framework* de desenvolvimento para apoiar a metodologia de desenvolvimento proposta, fornecendo um conjunto de linguagens de modelagem e técnicas de automação.

S2 (XIE *et al.*, 2015) descreve uma metodologia formal criada para desenvolver ‘ingredientes’ de sensores e atuadores, chamada FMDA (*Formal Methodology for Developing a Sensing/Actuating*), bem como um mecanismo para conectar todos esses ingredientes, chamado FMCA. A construção do FMDA foi baseada na arquitetura PMDA (*Physical Model Driven Software Architecture*) (XIE; CHEN; CUI, 2013) e foi elaborada em duas etapas: 1) transformação dos componentes da arquitetura PMDA em módulos de software; e 2) análise das relações entre esses módulos de software para que eles se conectem e formem um sistema de software composto de ‘ingredientes’ de sensores e atuadores. Depois de gerar os módulos de software, é criado um mecanismo chamado FMCA que permite a conexão entre dois ‘ingredientes’.

O estudo S3 (PRAMUDIANTO; INDRA; JARKE, 2013) propõe uma visão arquitetural para o desenvolvimento de protótipos IoT que tem como foco a separação da modelagem de domínio e das implementações tecnológicas. Além da arquitetura, o estudo também elabora uma ferramenta de Desenvolvimento Dirigida a Modelos para facilitar a construção de modelos de domínio de forma mais fácil, mapeando objetos virtuais e vinculando com as tecnologias de desenvolvimento. Por fim, a ferramenta é capaz de gerar código Java que possa ser estendido para desenvolver aplicações com mais detalhamento.

S4 (COSTA; PIRES; DELICATO, 2016) apresenta uma metodologia de engenharia de sistemas baseada em modelos para o desenvolvimento de aplicações de IoT, denominada IDeA - *IoT DevProcess & AppFramework*. IDeA é baseada na metodologia MBSE (*Model-Based Systems Engineering*) e utiliza padrões bem definidos, tais como o ISO/IEC/IEEE 15288, de

2015, e o OOSEM (*Object-Oriented Systems Engineering Standards*).

S5 (SINHA; NARULA; GRUNDY, 2017) apresenta um método chamado “Statechart paramétrico”, uma extensão de diagramas de estados para o desenvolvimento de aplicações de sistemas IoT. Os autores propuseram o método para tornar o projeto de aplicações mais flexível, considerando as configurações dinâmicas que acontecem em vários sistemas. Por exemplo, em s IoT, existem ambientes com vários sensores e atuadores e suas configurações podem mudar a qualquer momento.

S6 (AYALA; AMOR; FUENTES, 2014) propõe uma solução que faz o uso da *Common Variability Language* (CVL) (HAUGEN *et al.*, 2008), uma linguagem independente de domínio para especificar e resolver a variabilidade de qualquer instância de qualquer linguagem usando o metamodelo baseado em MOF (por exemplo, a UML e DSL), para o desenvolvimento de um processo de desenvolvimento de Sistemas Multi-Agente (*Multi Agent System*). Esse processo foi realizado com base nos agentes “Self-StarMAS”, pois segundo os autores, essa tecnologia de agentes pode ser incorporada em dispositivos IoT, tais como sensores, atuadores, dentre outros.

Por fim, o estudo S7 (FORTINO *et al.*, 2015) relata uma abordagem baseada em meta-modelo para o desenvolvimento sistemático de objetos inteligentes. O estudo explora o uso de diferentes metamodelos existentes para adaptá-los à metodologia de desenvolvimento de objetos inteligentes, sendo eles: i) Metamodelo de Objetos Inteligentes de Alto Nível (FORTINO *et al.*, 2013); ii) Metamodelo de Objeto Inteligente baseado em ELDA (FORTINO *et al.*, 2010); iii) Metamodelo *Smart Object* baseado em ACOS; e iv) Metamodelo JACOSO (BELLIFEMINE; POGGI; RIMASSA, 2001);

RQ 1.1: *Dentre as soluções, há alguma em que se aborda o conceito de Sistema-de-Sistemas (SoS)?*

No âmbito de IoT, não foram encontrados estudos que fazem o relacionamento entre o desenvolvimento de aplicações IoT de acordo com conceitos de SoS. Porém, como descrito no Capítulo 1, Maia *et al.* (2014) apresentou uma visão geral de cenários e abordagens atuais no desenvolvimento de SoS baseados em IoT, o conceito inverso em relação à pesquisa realizada pelo proponente desta dissertação. Os autores afirmam que, com a adoção do paradigma IoT, as “coisas” inteligentes estarão envolvidas em relações complexas, colaborando com outros sistemas independentes e heterogêneos para fornecer novas funcionalidades, formando um SoS de grande escala, como por exemplo, a composição de vários sistemas presentes em uma cidade inteligente.

Nesse sentido, a falta de estudos que abordem o desenvolvimento de aplicações IoT com base em SoS se justifica pelo fato de que ambas as áreas são consideradas novas no meio acadêmico e industrial. É importante ressaltar que o proponente deste trabalho, ao testar e refinar combinações de *strings* de busca, acrescentou o termo “Sistema-de-Sistemas” e seus sinônimos para encontrar estudos relacionados ao desenvolvimento de IoT e SoS, não obtendo o resultado esperado.

RQ 2: *Quais fases do ciclo de vida do desenvolvimento software o tipo de solução identificada aborda?*

Alguns estudos selecionados para extração de dados relataram suporte para uma ou mais fases no desenvolvimento de aplicações IoT, enquanto outros levantaram estudos sobre abstrações de alto nível para o desenvolvimento de aplicações IoT, recursos importantes para aplicações e comparações entre métodos para o desenvolvimento de aplicações que pode ser usado no paradigma da IoT, não explicitando as fases que apoiam no processo de desenvolvimento. A Tabela 9 mapeia os estudos para as fases em que eles se aplicam. Os estudos que não forneceram uma solução para as fases do ciclo de vida são S2 (PREHOFER, 2015), S3 (ZAMBONELLI, 2017) e S8 (DE *et al.*, 2011).

Tabela 9 – Fases de desenvolvimento dos estudos selecionados

	Análise	Projeto	Desenvolvimento
S1	x	x	x
S2	-	x	-
S3	-	x	x
S4	-	x	-
S5	-	x	x
S6	-	x	-
S7	-	x	x

Fonte: Elaborada pelo autor.

No estudo S1 (PATEL; CASSOU, 2015), a solução apoia várias fases na construção de uma aplicação IoT, começando na fase de projeto com a separação de interesses de acordo com a capacidade de cada *stakeholder* e também apoiando a fase de implementação através do uso de uma estrutura que oferece um conjunto de linguagens para desenvolvimento e, posteriormente, a geração de código.

A solução proposta pelo estudo S2 (XIE *et al.*, 2015) apoia a fase de projeto com a criação de uma metodologia para o desenvolvimento de ingredientes sensores e atuadores. O estudo S4 (COSTA; PIRES; DELICATO, 2016) também apoia somente a fase de projeto no desenvolvimento de aplicações IoT por meio da elaboração de uma abstração de desenvolvimento de aplicações levando em consideração o modelo de arquitetura proposto por Bassi *et al.* (2013).

O estudo S3 (PRAMUDIANTO; INDRA; JARKE, 2013) dá suporte às fases de projeto e desenvolvimento, pois a solução propõe uma visão arquitetural com foco na modelagem de domínio nos estágios iniciais de desenvolvimento e a geração de código a partir do modelo de domínio gerado. No estudo S5 (SINHA; NARULA; GRUNDY, 2017), a metodologia proposta é focada nas fases de projeto e desenvolvimento, pois os autores propuseram uma ferramenta capaz de gerar código em Java para aplicações móveis a partir dos modelos gerados. S6 (SINHA; NARULA; GRUNDY, 2017) apoia a fase de projeto por usar a CVL para o desenvolvimento de aplicações multi-agentes.

Por fim, no estudo S7 (FORTINO *et al.*, 2015), uma abordagem é proposta para o desenvolvimento de objetos inteligentes em alto nível, apoiando as fases de análise, projeto e desenvolvimento. Cada metamodelo explorado para o desenvolvimento de objetos inteligentes aborda uma fase. Na fase de análise, o Metamodelo de Objeto Inteligente de Alto Nível (FORTINO *et al.*, 2013) é explorado. Na fase de projeto, dois metamodelos são explorados: i) Metamodelo de objeto inteligente baseado em ELDA e Metamodelo de objeto inteligente baseado em ACOSO (FORTINO *et al.*, 2010). Finalmente, na fase de desenvolvimento, o metamodelo *JADE-based ACOSO* (BELLIFEMINE; POGGI; RIMASSA, 2001) é usado.

3.3 Considerações Finais

Neste capítulo, apresentou-se um mapeamento sistemático da literatura, que teve por objetivo a identificação e análise de estudos que apoiam o desenvolvimento de aplicações IoT. O desenvolvimento do MS foi executado de acordo com três etapas, sendo elas: planejamento, condução e sumarização.

Por meio desse MS, pôde-se verificar que existe uma carência de abordagens de apoio ao desenvolvimento de aplicações para IoT. Dentre as abordagens existentes, poucas auxiliam em todas as fases de desenvolvimento da aplicação. Além disso, de acordo com a análise realizada, foi possível observar que nenhum estudo analisado fez o uso de conceitos de SoS no desenvolvimento de aplicações desse tipo.

Outro ponto importante foi que, nos estudos encontrados e analisados nesse MS, os autores não realizaram aplicações das soluções no mundo real, desenvolvendo apenas provas de conceito de acordo com as propostas desenvolvidas.

A partir dessa carência, no próximo capítulo é apresentada a abordagem proposta pelo proponente deste trabalho de mestrado para o desenvolvimento de aplicações para IoT com base em conceitos de SoS. Essa abordagem foi baseada em abstrações de conceitos sobre o desenvolvimento de aplicações IoT de acordo com estudos encontrados na literatura

UMA ABORDAGEM PARA AUXILIAR O DESENVOLVIMENTO DE APLICAÇÕES PARA INTERNET DAS COISAS UTILIZANDO CONCEITOS DE SISTEMA-DE-SISTEMAS

4.1 Considerações iniciais

Conforme os resultados do MS apresentado no Capítulo 3, existe uma carência de abordagens para o desenvolvimento de aplicações IoT e, mesmo as existentes, não exploram os benefícios que poderiam ser alcançados ao se basear nos conceitos de SoS durante o desenvolvimento desse tipo de aplicação. Em cidades inteligentes, por exemplo, é possível encontrar diversas aplicações heterogêneas que poderiam ser integradas para formar um SoS, obtendo-se comportamentos maiores e bem definidos (comportamento emergente).

Dessa forma, neste capítulo é apresentada uma abordagem para desenvolvimento de aplicações IoT com base em conceitos de SoS. Essa abordagem constitui-se na elaboração de um guia que direcione o desenvolvedor a criar aplicações IoT, auxiliando nas principais fases de desenvolvimento. Os passos foram elaborados de acordo com as especificações de desenvolvimento de software orientado a objetos já consolidadas, como proposto por [Larman \(2007\)](#), [Pressman e Maxim \(2016\)](#) e [Sommerville \(2011\)](#), pela análise de estudos existentes na literatura que auxiliam o desenvolvimento de aplicações IoT, como visto no Capítulo 3, pelo de uso de conceitos de SoS, como descrito na Seção 2.3, e também por meio da utilização da arquitetura orientada a serviços, como descrito na Seção 2.4 para a integração entre sistemas heterogêneos presentes em um SoS.

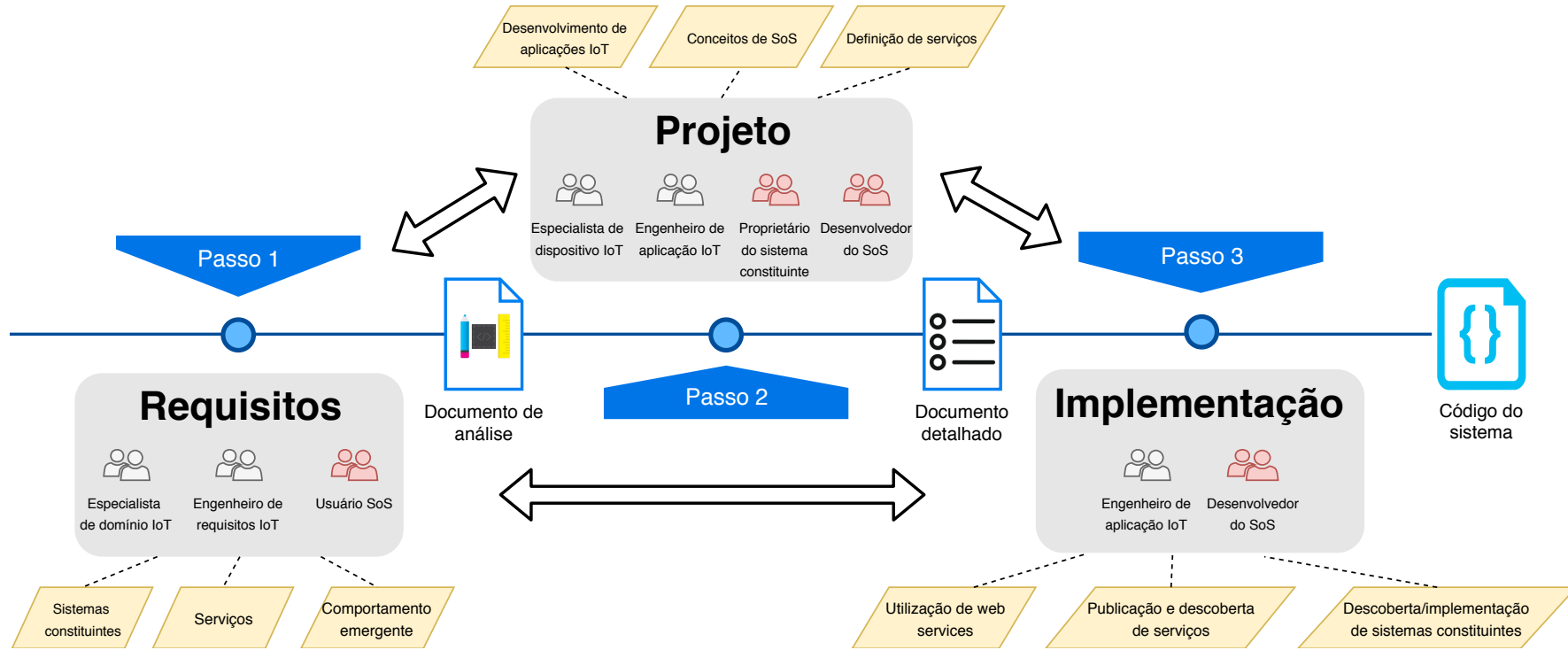
O restante deste capítulo está organizado da seguinte forma: na Seção 4.2, é apresentado uma visão geral da abordagem. Na Seção 4.3 é descrito o primeiro passo da abordagem, que

consiste na análise das informações iniciais da aplicação que se deseja desenvolver rumo a um SoS. Na Seção 4.4, é descrita a fase de projeto, contendo as atividades para elaboração de modelos com maior detalhamento da aplicação que será desenvolvida. Na Seção 4.5, é apresentado as atividades para implementação da aplicação, utilizando o paradigma orientado a serviços para promover a integração entre aplicações. Por fim, na Seção 4.7, são apresentadas as considerações finais deste capítulo.

4.2 Visão Geral da abordagem

A abordagem proposta visa auxiliar o desenvolvimento de aplicações contendo sistemas do domínio IoT ou não, cuja integração entre diversos sistemas possa alcançar um comportamento maior não previsto em qualquer um dos sistemas de forma isolada, tornando-se um SoS. Entende-se por abordagem um guia que auxilie na elaboração desse tipo de aplicações IoT rumo a um SoS, cobrindo as fases de requisitos, projeto e implementação, com destaque na fase de projeto, na qual são feitas associações dos conceitos de IoT e SoS. Em cada passo, são propostas atividades a serem realizadas por diferentes especialistas para que o entendimento e desenvolvimento da aplicação se torne fácil de ser realizada. A abordagem proposta por este trabalho é incremental e iterativa, ou seja, as atividades podem passar por diversas iterações, produzindo em cada uma delas um incremento em que porções da funcionalidade vão sendo incorporadas até chegar ao SoS completo. Na Figura 15 é ilustrada a abordagem proposta.

Figura 15 – Abordagem proposta pelo proponente

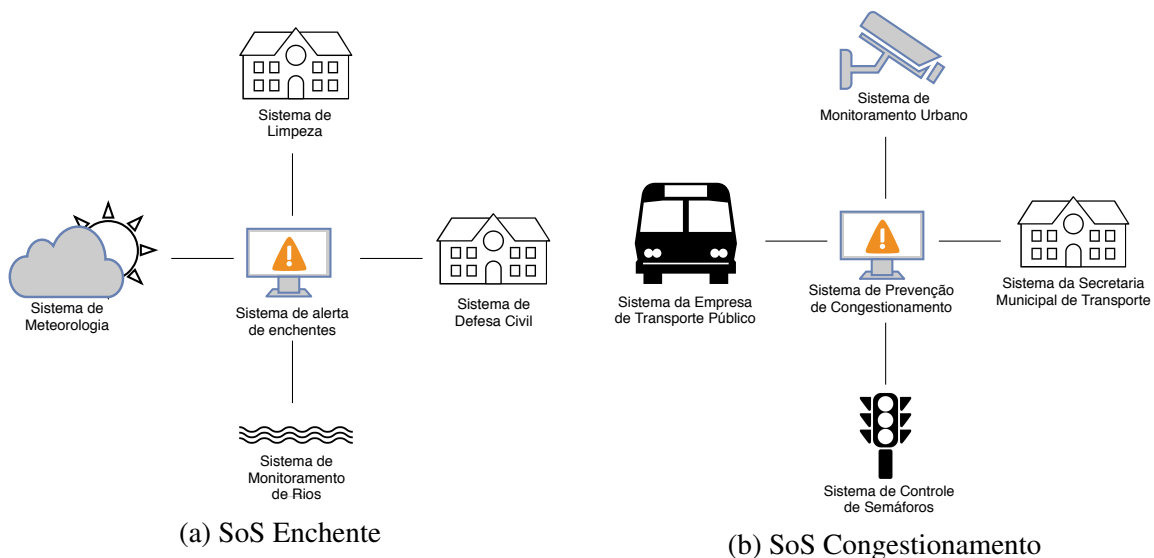


Fonte: Elaborada pelo autor.

O público alvo da abordagem deste trabalho são desenvolvedores que desejam implementar um sistema que consista na utilização das funcionalidades de outros sistemas, existentes ou não, sendo que um ou mais desses sistemas se encaixam no paradigma IoT. Outro ponto importante é que a integração de diversos sistemas heterogêneos existentes podem resultar em comportamentos maiores do que em apenas um sistema, levando ao conceito de SoS. Essa integração pode ser feita por meio da utilização da arquitetura orientada a serviços, pois essa arquitetura oferece benefícios como baixo acoplamento e interfaces bem definidas. Os sistemas, denominados de SCs, são operacionalmente e gerencialmente independentes, colaborando entre si e fornecendo serviços para um ou mais SoS.

Exemplos de desenvolvimento de aplicações que envolvem sistemas do domínio IoT e de outros domínios podem ser elaborados por meio da utilização dessa abordagem, rumo ao desenvolvimento de SoS, como ilustra a Figura 16. A integração entre sistema de meteorologia e sensores que monitoram rios, por exemplo, pode se transformar em apoio para alertar entidades responsáveis a tomarem iniciativas no combate a possíveis catástrofes (Figura 16a). Outro exemplo é a integração entre sistema de monitoramento por câmeras e sistemas de controle de semáforos que, ao detectar grandes volumes de veículos, podem alertar outros sistemas para realocar o fluxo e evitar grandes volumes de congestionamento (Figura 16b).

Figura 16 – Exemplos de integração de aplicações IoT e de outros domínios



Fonte: Elaborada pelo autor.

A seguir, são descritos os passos da abordagem necessários para o desenvolvimento de aplicações IoT para se tornar um SoS, de acordo com os papéis dos diferentes *stakeholders* envolvidos nas diferentes fases de desenvolvimento.

4.3 Passo 1: Requisitos

Este passo tem por objetivo a obtenção das informações iniciais do SoS, ou seja, o planejamento do comportamento emergente esperado de acordo com as funcionalidades oferecidas por cada SC. Essa fase não leva em consideração detalhes técnicos como tipos de dispositivos, tecnologias ou plataformas. Este passo procura responder, por meio da atividade “**Obtenção das principais informações da aplicação**”, algumas perguntas para a obtenção das informações iniciais na fase de requisitos no desenvolvimento de uma aplicação IoT, sendo elas:

1. Qual é o comportamento emergente desejado?
2. Quais são os SCs necessários para que aconteça o comportamento emergente?
3. Quais dos sistemas são do domínio IoT, ou seja, em quais desses existem sensores e/ou atuadores que monitoram e/ou realizam alguma ação em determinados locais?
4. Quais são os *stakeholders* presentes nesse passo e quais são suas responsabilidades?

Atividade: Obtenção das principais informações do SoS

Responsáveis: Especialistas de domínio IoT, engenheiro de requisitos IoT e usuário do SoS.

Entradas: Conhecimento do domínio de cada responsável e requisitos do usuário.

Saídas: Modelos de representação em alto nível do sistema.

Objetivo: Definir as principais funcionalidades do SoS, identificando como cada SC pode contribuir para atingir o comportamento emergente.

Descrição: Este passo compreende o levantamento das informações preliminares para o desenvolvimento do SoS. A obtenção dessas informações permite a visualização do sistema de uma forma mais ampla. Os *stakeholders* (Tabela 10) devem identificar quais os principais SCs envolvidos e quais os serviços que eles podem oferecer para que o comportamento emergente seja atingido.

Tabela 10 – Papéis dos *stakeholders* no passo de análise

Responsável	Papel	Referência
Especialista de domínio IoT	Definir os requisitos e o escopo da aplicação	Costa, Pires e Delicato (2016)
Engenheiro de requisitos IoT	Elicitação dos requisitos dos dispositivos IoT	Costa, Pires e Delicato (2016)
Usuário SoS	Definir as capacidades para atingir o comportamento emergente	Lewis <i>et al.</i> (2011)

Fonte: Elaborada pelo autor.

Deve-se ressaltar que nem sempre os serviços e recursos estão disponíveis em sistemas existentes e, portanto, pode ser necessário desenvolver tais serviços ou novos constituintes que os ofereçam. Os *stakeholders* também devem analisar qual é a missão de cada SC de forma individual e qual será a missão do SoS como um todo a partir dos recursos oferecidos por cada SC.

O artefato de saída para essa atividade é um documento contendo as informações em alto nível do SoS a ser projetado. Assim, para a obtenção desse artefato, este trabalho recomenda a utilização de casos de uso, que são narrativas em texto cujo objetivo é descrever as principais funcionalidades do sistema (LARMAN, 2007). Casos de uso podem ser descritos de três formas diferentes. Porém, a forma utilizada para o desenvolvimento de aplicações IoT é a resumida, onde um breve resumo de apenas um parágrafo descreve o cenário de sucesso principal do sistema. No caso de desenvolvimento de aplicações IoT rumo a um SoS, o sucesso principal é atingir o comportamento emergente por meio da identificação e integração entre os SCs.

É importante que, na descrição do caso de uso, os *stakeholders* deixem claro qual o objetivo do SoS, descrevendo quais são os SCs que serão necessários para que o comportamento emergente seja obtido. Além disso, como o objetivo é desenvolver uma aplicação que tenha conceitos de SoS, deve-se deixar claro a utilização de serviços para a integração entre os diferentes SCs que serão necessários para atingir o objetivo do SoS (comportamento emergente). Outro ponto importante é deixar claro qual a missão de cada SC que estará presente no SoS, pois cada um trabalha de forma individual e independente.

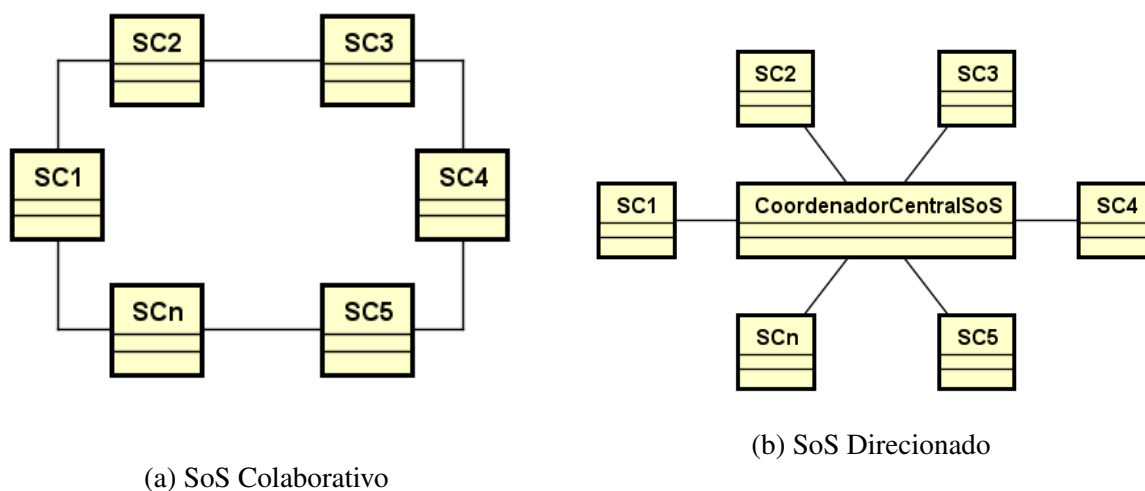
Além da descrição do caso de uso, a abordagem recomenda o uso de diagramas da UML para auxiliar os *stakeholders*. Portanto, indica-se a utilização do diagrama de casos de uso, pois auxilia na identificação das principais funcionalidades e suas interações, além de ser intuitivo e facilmente compreendido por *stakeholders* que não são da área de computação.

Além do diagrama de casos de uso, outros diagramas podem auxiliar a representação do sistema em níveis mais altos de abstração. Dessa forma, recomenda-se a utilização do diagrama de classes de forma mais simples para a representação conceitual do sistema, ilustrando as principais ligações entre os diferentes SCs presentes no SoS, como mostra a Figura 17.

Recomenda-se também a utilização do diagrama de sequência para modelar as interações entre os SCs que farão parte do SoS. Por meio da utilização do diagrama de sequência, é possível identificar se o SoS seguirá uma estratégia proativa ou reativa. Na estratégia proativa (Figura 18), o comportamento emergente é obtido por meio da ação do coordenador central de forma antecipada, ou seja, o objetivo do SoS é definido e a autoridade central busca pelo comportamento emergente com a obtenção de dados, informações ou recursos dos SCs.

Na estratégia reativa (Figura 19), o comportamento emergente é atingido por meio da interação de outros SCs com o coordenador central, ou seja, um ou mais constituintes avisam o coordenador central sobre algum comportamento fora do padrão para que a obtenção do

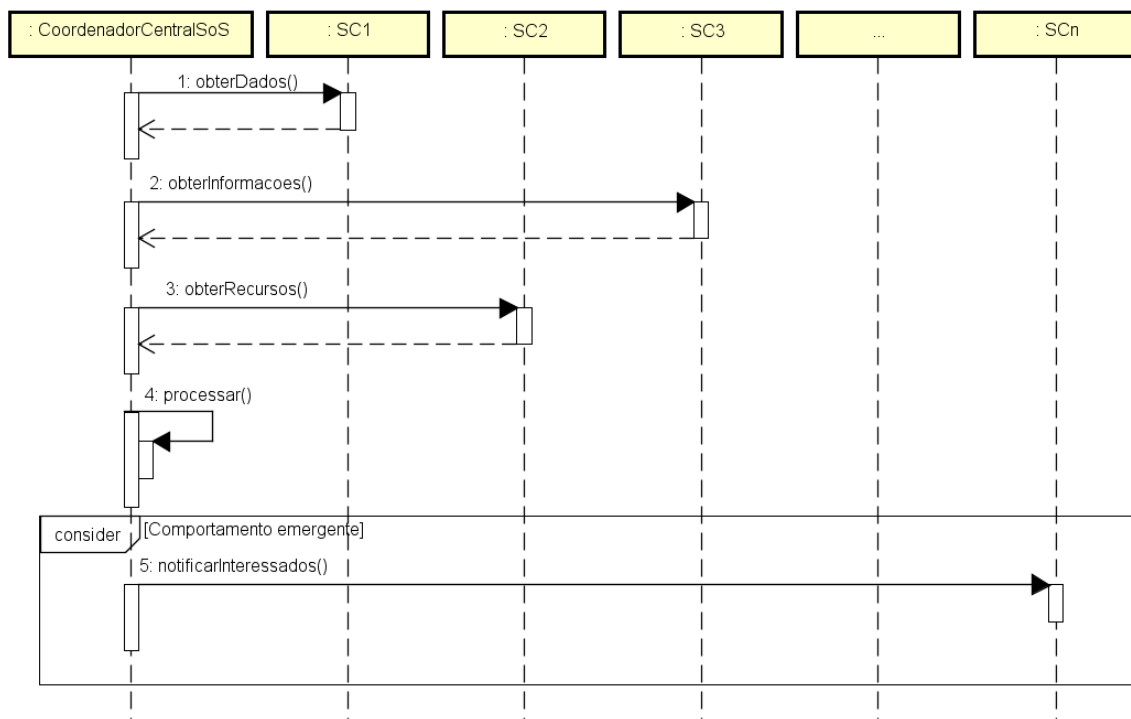
Figura 17 – Representação alto nível utilizando diagrama de classes



Fonte: Elaborada pelo autor.

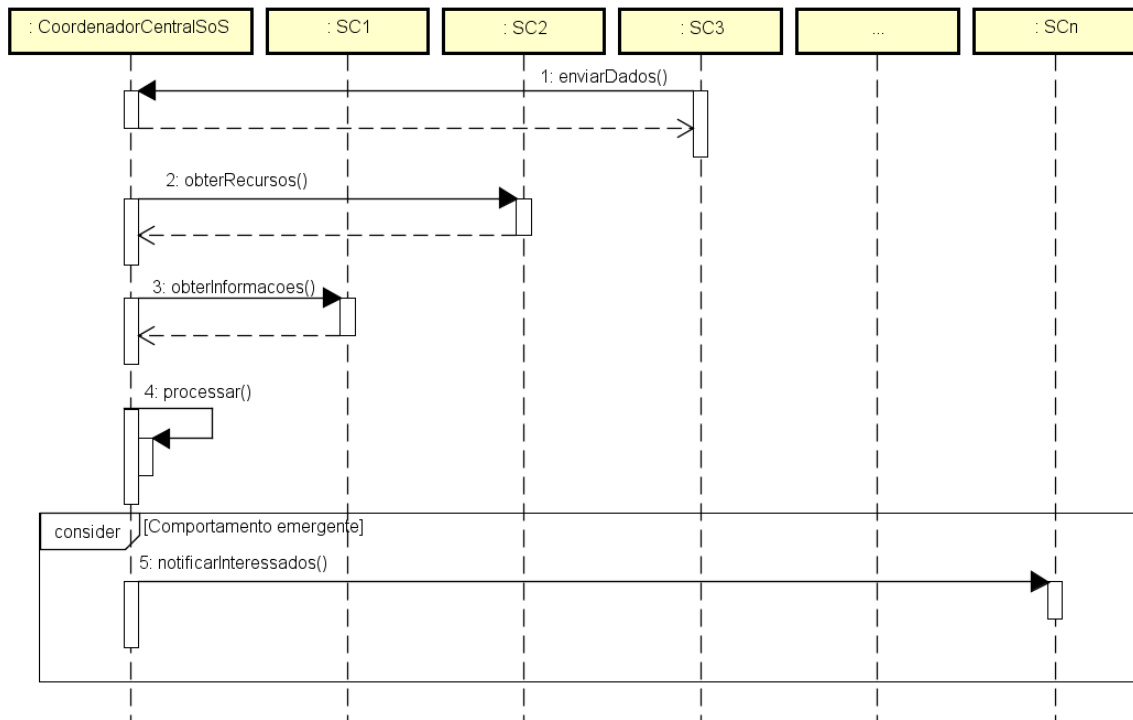
comportamento emergente seja disparada pelo coordenador. O coordenador central faz a busca de dados, informações ou recursos de outros SCs envolvidos para atingir o objetivo. É indicado ressaltar no diagrama de sequência, como são ilustrados nas Figuras 18 e 19, o comportamento emergente que é esperado pelo SoS.

Figura 18 – Representação alto nível utilizando diagrama de sequência - SoS Proativo



Fonte: Elaborada pelo autor.

Figura 19 – Representação alto nível utilizando diagrama de sequência - SoS Reativo



Fonte: Elaborada pelo autor.

4.4 Passo 2: Projeto

Esse passo compreende a elaboração de modelos detalhados do sistema, especificando os conceitos dos domínios de IoT a partir do documento gerado na fase de requisito. Esse passo tem como intuito, por meio da atividade “**Projetar o SoS**”, responder às seguintes perguntas com mais detalhes:

1. Quais são os SCs do domínio IoT?
2. Quais são os SCs fora do domínio IoT?
3. Quem são os *stakeholders* responsáveis nesse passo?
4. Quais são os serviços que os SCs disponibilizam para atingir o comportamento emergente do SoS?

Atividade: Projetar o SoS

Responsáveis: Especialista de dispositivo IoT, engenheiro de aplicação IoT e proprietário do SC.

Entradas: Documento de requisitos gerado na fase de requisito, conhecimento no domínio IoT e SoS.

Saídas: Modelos com mais detalhes do SoS.

Objetivo: Definir modelos detalhados do sistema que permitam a posterior implementação.

Descrição: O modelo de representação em alto nível do sistema foi elaborado no passo 1 usando linguagem natural, para facilitar a comunicação e o entendimento dos principais conceitos envolvidos para o desenvolvimento do SoS envolvendo diversos SCs, sendo ou não do domínio IoT. Além disso, foi sugerido criar diagramas de visão geral do SoS. A partir desse documento textual e dos diagramas criados, um modelo mais detalhado deve ser obtido descrevendo as diferentes abstrações dos diferentes domínios, identificando detalhes específicos que irão ajudar na fase de implementação do SoS, tais como o recurso que será oferecido por um determinado serviço, onde os serviços estarão armazenados e quais os sistemas envolvidos em cada serviço.

Para a realização dessa fase, este trabalho propõe a utilização das abstrações realizadas pelo proponente para geração de modelos com maiores níveis de detalhes (Figura 20). Tais abstrações foram adaptadas do estudo de [Costa, Pires e Delicato \(2016\)](#), que utilizou conceitos do modelo de domínio IoT proposto por [Bassi et al. \(2013\)](#) com base no processo de desenvolvimento de sistemas complexos de acordo com a ISO/IEC/IEEE 1528 de 2015, como descrito na Seção 2.5. Além dessas abstrações, também foram consideradas as definições e conceitos de SoS descritos na Seção 2.3 e as ontologias elaboradas por [Abdalla \(2017\)](#), que formaliza termos e também conceitos de SoS. Esse modelo permite a elaboração de modelos para o domínio IoT e também para sistemas convencionais.

O modelo conceitual proposto para a fase de projeto é apresentado na Figura 20, destacando-se os conceitos de SoS e os aspectos relacionados com o domínio IoT, enfatizando que aplicações desse domínio são dispostos de sensoriamento e/ou atuação ([COETZEE; EKSTEEN, 2011](#)). Os detalhes do modelo conceitual são descritos a seguir.

Um **Sistema Constituinte** é uma composição de **Dispositivos** e **Serviços** que pode interagir com outros **Sistemas Constituintes** e também com **Entidades Físicas**. Um **Sistema Constituinte** pode ser tanto do domínio IoT quando fora desse domínio. Um **Dispositivo**, um artefato técnico destinado a fornecer uma interface entre os mundos digital e físico, como por exemplo um Arduino, pode ser um **Sensor**, que monitora uma **Entidade Física**, ou um **Atuador** que realiza alguma ação em uma **Entidade Física**. Porém, um **Dispositivo** também pode ser um sistema de informação sem a necessidade de ser um Sensor ou um Atuador. Uma **Entidade Física** é representada por uma **Entidade Virtual**, e essa relação é alcançada por meio da utilização de dispositivos que fornecem uma interface tecnológica para interagir ou obter informações sobre a entidade física. Esses dispositivos podem ser **Sensores** e/ou **Atuadores**.

Um **Sensor** é responsável por monitorar um determinado ambiente (**Entidade Física**), coletando informações em tempo real. Essa entidade é composta por três atributos: *sensorID*, especificando de forma única o sensor; *latitude* e *longitude*, especificando a posição geográfica. Um **Atuador** é responsável por executar alguma ação de acordo com as informações verificadas

pelo **Sensor**. Um **Serviço**, que expõe zero ou mais **Recursos**, possui quatro atributos: o *serviceType*, que especifica a tecnologia usada para invocá-lo (como, por exemplo, REST); *in e out*, definindo tipos de dados como parâmetros de entrada e resultados de saída; Um **Recurso**, por sua vez, possui três parâmetros: o *resourceID*, utilizado para identificá-lo exclusivamente, definido como *URI (Universal Resource Identifier)*; o *nome do recurso*; e, *resourceType*, que especifica o tipo do Recurso (dispositivo, sensor ou atuador). Um determinado **Recurso** que está disponível por meio de um **Serviço** pode estar hospedado tanto em um dispositivo físico local (**Recurso no dispositivo**) ou na rede (**Recurso na Rede**).

Em **System-of-Systems**, estão contidos os conceitos sobre SoS. Um **Sistema-de-Sistemas** é um conjunto de sistemas resultantes da integração entre **SistemasConstituintes** independentes e úteis em um sistema maior que oferece recursos únicos. SoS são categorizados (**Tipos-de-Sistemas-de-Sistemas**) de acordo com a extensão da independência gerencial apresentada pelos seus SCs, a saber: i) **Centralizado**, no qual o SoS é gerenciado por um **Coordenador Central**; e ii) **Descentralizado**, tipo de SoS em que não é gerenciado por uma autoridade central. No tipo **Centralizado**, existem dois tipos, sendo eles: i) **Direcionado** e ii) **Reconhecido**. No tipo **Descentralizado** também existem dois tipos: i) **Virtual** e **Colaborativo**.

Com base no modelo da Figura 20, os responsáveis devem analisar as principais informações obtidas do SoS, que envolvem conceitos do domínio IoT e de outros domínios, e instanciar cada elemento do modelo conceitual proposto na fase de projeto, atribuindo uma ou mais instâncias a cada conceito (ou SC). Os responsáveis também devem analisar quais são os recursos oferecidos, por meio de serviços, de cada SC para que o objetivo do SoS seja atingido e quais os dispositivos envolvidos, sendo eles sistemas convencionais ou do domínio IoT. Para especificar os serviços, sugere-se que os *stakeholders* descrevam qual o tipo de tecnologia que será utilizada (por exemplo, REST ou SOAP), o endereço (caminho) que estarão disponíveis os recursos de cada serviço e onde estarão disponíveis (localmente ou disponíveis na rede), a descrição do serviço, os parâmetros de entrada e os parâmetros de saída, ou seja, o que o serviço irá retornar para quem o invoca.

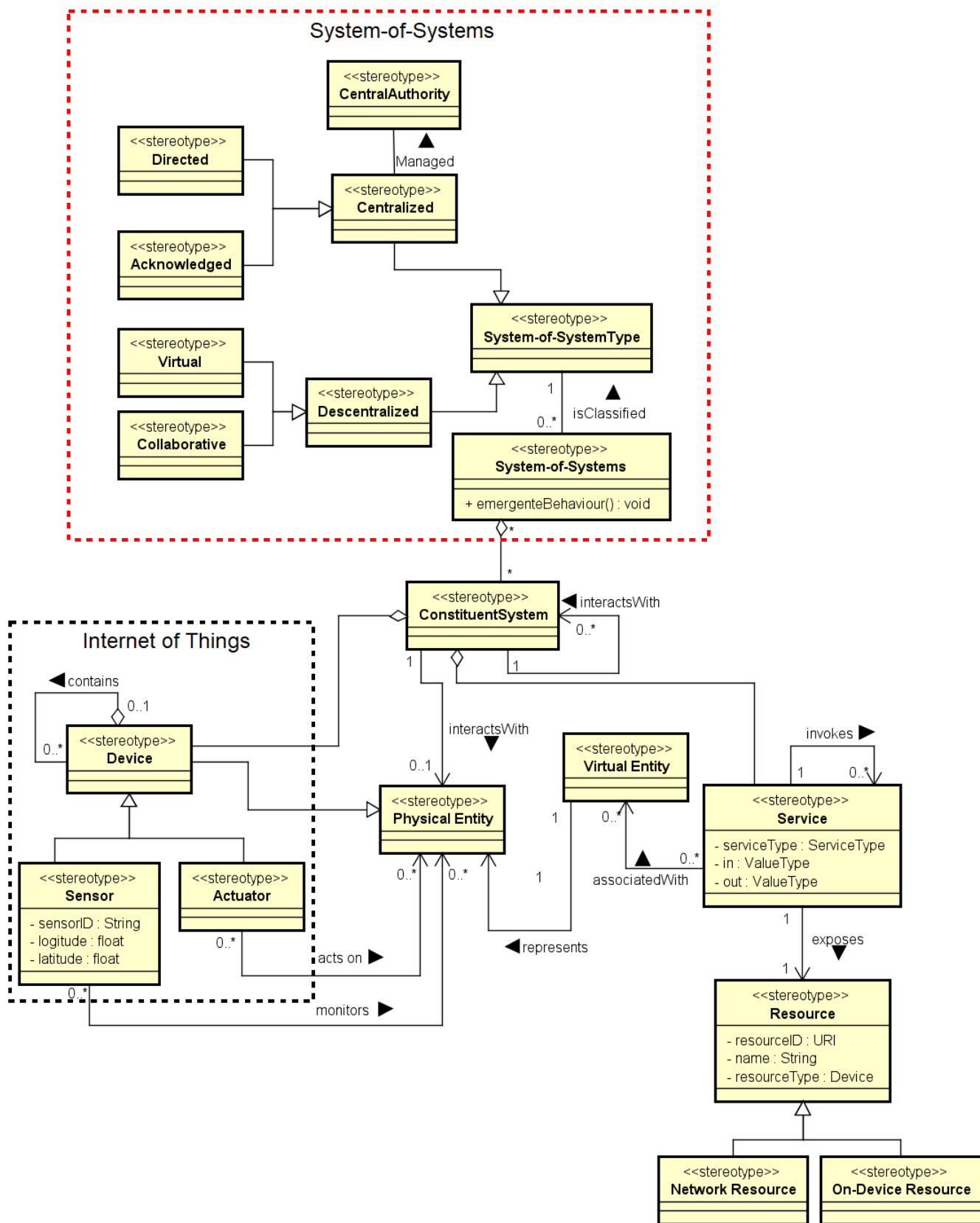
Os papéis dos responsáveis durante a fase de projeto de aplicações IoT com base em SoS são descritos na Tabela 11.

4.5 Passo 3: Implementação

Esse passo tem por objetivo a codificação dos sistemas a partir do modelo de projeto detalhado realizado na fase de projeto, com intuito de responder às seguintes perguntas por meio da atividade “**Implementação do SoS**”:

1. Qual tecnologia usar para implementar serviços?
2. Como especificar as interfaces dos serviços?

Figura 20 – Modelo conceitual da abordagem proposta na fase de projeto



Fonte: Elaborada pelo autor.

Tabela 11 – Papéis dos *stakeholders* na fase de projeto

Responsável	Papel	Referência
Especialista de dispositivo IoT	Especificar os serviços que os dispositivos irão disponibilizar	Costa, Pires e Delicato (2016)
Engenheiro de aplicação IoT	Especificar a estrutura e comportamento da aplicação	Costa, Pires e Delicato (2016)
Proprietário do sistema constituinte	Especificar e/ou desenvolver constituintes de acordo com a necessidade do SoS	Lewis <i>et al.</i> (2011)
Desenvolvedor do SoS	Especificar instâncias do projeto de acordo com os SCs do SoS	Lewis <i>et al.</i> (2011)

Fonte: Elaborada pelo autor.

3. Onde deixar os serviços disponíveis?
4. Quais são os responsáveis por essa fase?

Atividade: Implementação do SoS

Responsáveis: Engenheiro de aplicação IoT e desenvolvedor do SoS

Entradas: Conhecimento dos *stakeholders* e modelo com maior nível de detalhes, elaborado na fase de projeto

Saídas: Codificação do SoS

Objetivo: Implementar o SoS e, caso não existam, os SCs que irão permitir atingir o comportamento emergente

Descrição: Com a elaboração do documento com um nível maior de detalhes na fase de projeto, os *stakeholders* responsáveis poderão iniciar o processo de codificação do SoS. Devido a integração entre diferente SCs, podendo ser ou não do domínio IoT, é necessário o desenvolvimento de aplicações utilizando a arquitetura orientada a serviços. É importante observar que, mesmo que os SCs existam, pode ser necessário implementar novos serviços para atender o que foi projetado pelo SoS. Caso existam outros SCs de outros proprietários e esses não ofereçam serviços, os responsáveis podem solicitar a implementação de uma camada de serviços ao proprietário para que esse SC faça parte do SoS.

Uma forma de implementar o paradigma de orientação a serviços é por meio de *web services*. Existem dois tipos de padrões de *web services* que podem ser utilizados para o desenvolvimento de aplicações baseadas no paradigma orientado a serviços, sendo eles: SOAP e RESTful (ver Seção 2.4). Sendo assim, para a realização dessa fase, esta dissertação propõe aos responsáveis (ver Tabela 12) a utilização de *web services* do tipo RESTful, pois além de oferecer maior flexibilidade ao desenvolvedor na escolha do formato das mensagens trocadas,

os serviços REST são mais leves e mais rápidos. Outro ponto importante é disponibilização de documentação. Essa é apenas uma recomendação para desenvolvedores não familiares com o paradigma de serviços, ficando os desenvolvedores experientes livres para escolherem o padrão que for mais conveniente para seu contexto.

Tabela 12 – Papéis dos *stakeholders* na fase de implementação

Responsável	Papel	Referência
Engenheiro de aplicação IoT	Especificar a estrutura e comportamento da aplicação	Costa, Pires e Delicato (2016)
Desenvolvedor do SoS	Desenvolver e gerenciar o SoS	Lewis <i>et al.</i> (2011)

Fonte: Elaborada pelo autor.

Com a recomendação da utilização do REST, os responsáveis devem considerar os aspectos de implementação desse padrão para especificar a interface de acesso ao serviço e para manipular os recursos disponíveis. Os recursos podem ser acessados ou manipulados por meio de um conjunto de operações predefinidos pelo padrão, ou seja, utilizando os verbos HTTP como anotações, sendo que cada método HTTP vem precedido pelo símbolo @. A interface de acesso é feita por meio da definição de URI. Cada serviço possui uma única identificação para ser acessado, ou seja, possui uma única URI. A Tabela 13 contém as anotações do padrão REST para implementação de serviços.

Tabela 13 – Anotações REST

Anotação	Descrição
@Path	Determina o caminho relativo no qual o serviço será hospedado no servidor
@Get	Obtenção de recurso do serviço
@Post	Criação de recurso
@Put	Atualização de recurso
@Delete	Remover recurso
@PathParam	Designação de parâmetro na URI para requisição do serviço
@QueryParam	Designação de parâmetro de busca pela URI para atualizar ou modificar serviço
@Consumes	Especificar o tipo de dado que um recurso pode consumir
@Produces	Especificar o tipo de dados que um recurso pode produzir e enviar para o cliente em resposta a uma solicitação

Fonte: Elaborada pelo autor.

Um exemplo de como utilizar o padrão REST é ilustrado pelo Código-fonte 1. Nesse exemplo, o caminho relativo, determinado pela anotação @**Path**, para representar a classe Exemplo é definido como “servico”. A anotação @**GET** indica que o método irá processar requisições para fornecer recursos a quem acessa. A anotação @**Path** abaixo do @GET determina o caminho

relativo do método `olaMundo()`, que está hospedado no caminho `"/servico"`. Dessa forma, para acessar o método `olaMundo()`, é necessário acessar o método pelo caminho `"/servicos/oi"`. Por fim, a anotação `@Produces` determina o tipo de dado produzido pelo método. Nesse exemplo, o tipo de dado que será retornado ao cliente será do tipo texto puro.

Código-fonte 1 – Exemplo do padrão REST

```
1:     @Path("servico")
2:     public class Exemplo
3:     {
4:         @GET
5:         @Path("ola")
6:         @Produces(MediaType.TEXT_PLAIN)
7:         public String olaMundo()
8:         {
9:             return "Olá mundo!";
10:        }
11:    }
```

Os serviços implementados de uma aplicação são armazenados em um servidor, contendo a interface (ou caminho) de acesso para que seus recursos possam ser utilizados. É importante que os serviços sejam publicados em um repositório, contendo informações como descrição, parâmetros de entrada e de saída, e também o endereço de acesso para que outros SCs invoque-os e possam utilizar seus serviços.

Assim, com os serviços disponíveis, é possível que os SCs de um SoS possam integrar para atingir o comportamento emergente por meio da ação do coordenador central, como ilustrado na Figura 18, ou por meio da interação de outros SCs, que não sejam o coordenador central, mas que informam a ele sobre algo que está fora do funcionamento normal, como ilustrado na Figura 19. É importante ressaltar que em ambas as estratégias apresentadas pelas Figuras 18 e 19, o código que irá implementar o comportamento emergente será somente no coordenador central. A implementação desse código é de responsabilidade do engenheiro do SoS.

4.6 Discussão

Essa abordagem visa auxiliar o processo de desenvolvimento de aplicações IoT, de forma que haja colaboração entre vários sistemas e/ou “coisas”, permitindo que possam trabalhar juntos para prover um comportamento maior e mais definido, levando ao conceito de SoS. De acordo com [Cavalcante et al. \(2016\)](#), um dos desafios encontrados na formação de SoS em ambientes inteligentes, como em cidades inteligentes, é a integração entre diferentes projetos e aplicações heterogêneas encontradas nesse tipo de ambiente.

Levando em consideração esse aspecto, este trabalho propõe a utilização da arquitetura orientada a serviços para o desenvolvimento de aplicações IoT, pois essa arquitetura facilita a integração entre diferentes sistemas devido ao baixo acoplamento, padronização na linguagem e comunicação entre os envolvidos, suporta serviços independentemente da tecnologia utilizada e também permite a reutilização dos serviços.

A abordagem passo a passo proposta tem como propósito auxiliar os desenvolvedores a elaborar aplicações IoT rumo a um SoS nas diferentes fases do processo de desenvolvimento de software. O desenvolvimento de aplicações seguindo essa abordagem pode ser realizado de maneira iterativa e incremental, de forma que o SoS pode ser inicialmente desenvolvido com poucos SCs e, incrementalmente, ir acrescentando novos comportamentos até chegar ao SoS desejado. Além disso, os responsáveis podem interagir em outras fases do processo, colaborando melhor para que SoS seja desenvolvido com qualidade.

Deve-se observar que o processo de desenvolvimento contém outros passos que ficaram fora do escopo da abordagem proposta, como a validação do SoS, a implantação e futura manutenção/evolução do SoS. A definição desses passos fica como sugestão para trabalhos futuros.

4.7 Considerações finais

Neste capítulo foi apresentada a abordagem elaborada para o desenvolvimento de aplicações IoT rumo ao SoS, abordando a utilização de SOA como forma de promover a integração entre os diferentes SCs existentes. Também foram mostrados os passos conduzidos em cada fase no processo de desenvolvimento para a elaboração da proposta de desenvolvimento de uma aplicação com SCs do domínio IoT e também de outros domínios.

Com a utilização da abordagem, profissionais que desejam desenvolver aplicações, que contenham ou não sistemas da IoT, contam com um meio de conduzir e direcionar o processo de desenvolvimento, desde as fases iniciais até as fases finais, de forma que a integração entre as diferentes aplicações possam prover um comportamento maior não previsto em apenas uma aplicação, o comportamento emergente, levando ao desenvolvimento de um SoS.

Além disso, é possível ter em mãos uma documentação mais completa, permitindo assim que em futuras manutenções e/ou evoluções, esse processo seja facilitado por outros profissionais.

No Capítulo 5, é apresentada uma simulação realizada como prova de conceito para a abordagem proposta.

UMA PROVA DE CONCEITO DA ABORDAGEM

5.1 Considerações iniciais

Idealmente, a avaliação da abordagem proposta no Capítulo 4 deveria levar em consideração a análise de viabilidade de seu uso por especialistas da área de IoT e SoS e, em seguida, a análise de outros desenvolvedores ao criarem aplicações IoT seguindo os passos da abordagem, por meio de experimentos controlados. No entanto, essa avaliação demandaria não somente tempo, mas também recursos difíceis de obter no tempo de um mestrado, como por exemplo especialistas nas áreas já mencionadas e desenvolvedores fora do meio acadêmico. Assim, para viabilizar o término do projeto no tempo disponível, decidiu-se realizar uma simulação, feita pelo próprio mestrando, para servir como prova de conceito da abordagem. O objetivo do desenvolvimento dessa simulação é mostrar indícios de que, seguindo a abordagem proposta, é possível elaborar aplicações IoT rumo a um SoS, onde a integração de diversos sistemas, entre os quais alguns envolvendo coisas da IoT, pode gerar um comportamento emergente.

O objetivo deste capítulo é descrever essa simulação em detalhes. A simulação foi denominada “SoS Sai Dengue” e foi elaborada utilizando o processo de desenvolvimento apresentado no Capítulo 4. Exemplos dos artefatos produzidos ao longo de todo o processo são apresentados para ilustrar a abordagem.

Este capítulo está organizado da seguinte maneira: na Seção 5.2 encontra-se a descrição do exemplo utilizado, contendo aplicações convencionais e aplicações do domínio IoT. Na Seção 5.3, apresenta-se o processo de desenvolvimento da prova de conceito do exemplo utilizado, seguindo a abordagem proposta por este trabalho. Na Seção 5.4, apresenta-se o funcionamento do exemplo desenvolvido. Por fim, a Seção 5.5, as considerações finais deste capítulo são apresentadas.

5.2 Descrição do exemplo utilizado

Como exemplo para a prova de conceito, foi utilizada uma simulação baseada em um sistema concreto que poderá vir a ser desenvolvido em trabalhos futuros. O sistema envolve tanto sistemas convencionais quanto aplicações IoT e o objetivo é obter um SoS para prevenção a surtos de dengue. Como o desenvolvimento de um SoS deste porte não seria factível em um trabalho de mestrado, optou-se pela simulação.

Em uma pesquisa realizada, [Silva et al. \(2015\)](#) verificaram que a utilização de aprendizado de máquina permite que um determinado tipo de sensor identifique espécies de mosquito sem a intervenção humana. Os sensores capturam informações de voo de insetos por meio de laser de luz e os classificam de acordo com sua espécie. Essas informações podem ser transmitidas em tempo real, informando com exatidão a existência de vetores de doença em determinada região.

Considerando o sensor que captura informações de insetos voadores proposto por [Silva et al. \(2015\)](#), a simulação, que foi desenvolvida como prova de conceito da abordagem proposta pelo proponente deste trabalho, consiste em uma aplicação IoT que utilize sensores para verificar a existência do *Aedes Aegypti*, o mosquito responsável pela transmissão da dengue, *chikungunya* e zika vírus¹. A aplicação, denominada “SoS Sai Dengue”, tem como objetivo a obtenção das informações de presença do mosquito *Aedes* obtidas pelos sensores e informar entidades responsáveis para a realização de ações de combate e prevenção. Essa interação e integração entre as diferentes entidades permite que ações contra o mosquito *Aedes* sejam mais efetivas do que apenas a ação de uma única entidade.

Para manter o sistema pequeno a fim de não tornar complexa a prova de conceito, a simulação considera apenas o sistema que gerencia a rede de sensores, que captam informações do mosquito, os sistemas de duas entidades municipais que realizam as ações de combate e prevenção, e o sistema de alerta, que será responsável por prover o comportamento emergente com a integração entre os sistemas. A Figura 21 ilustra o cenário do SoS Sai Dengue.

É importante ressaltar que qualquer outro sistema que forneça informações relevantes para o SoS, sejam elas coletadas por sensores ou não, podem ser usados de maneira análoga, ou seja, eles podem ser utilizados na composição do SoS e contribuir para a obtenção de outros comportamentos emergentes.

5.3 O processo de desenvolvimento

Nas subseções seguintes são descritas as fases de requisitos, projeto e implementação da simulação, adotando a abordagem proposta para o exemplo da aplicação SoS Sai Dengue.

¹ <<http://combateaedes.saude.gov.br/pt/>>

Figura 21 – Cenário ilustrativo do SoS Sai Dengue



Fonte: Elaborada pelo autor.

5.3.1 Requisitos

De acordo com a abordagem proposta pelo proponente, para a fase de requisitos no desenvolvimento da simulação, utilizou-se casos de uso no formato resumido, descrevendo o objetivo do SoS a ser desenvolvido. O objetivo deve relatar o comportamento emergente que se deseja atingir. O caso de uso, denominado “Evitar Epidemia”, está descrito na Tabela 14.

Tabela 14 – Descrição de uso do cenário principal

<p>Caso de uso: Evitar Epidemia</p> <p>Visão geral: O Sistema de Alerta faz a consulta periodicamente de todos os dados gerados pelos sensores para realizar a análise. Caso seja detectada a presença do mosquito <i>Aedes</i>, o Sistema de Alerta envia notificação para que as entidades responsáveis tomem ações necessárias contra o inseto.</p>
--

Fonte: Elaborada pelo autor.

Por meio da descrição do caso de uso, é possível perceber que o SoS formado por diferentes SCs, tanto do domínio IoT quanto de outros domínios, será do tipo Direcionado, ou seja, um SoS do tipo centralizado em que o comportamento emergente (evitar epidemia) acontecerá por meio da presença de um coordenador central (sistema de alerta), responsável pela comunicação e integração entre as diferentes entidades responsáveis, consideradas como SCs. Para ilustrar esse cenário inicial, utilizou-se o diagrama de caso de uso (Figura 22), no qual o responsável pelo alerta, o “Sistema de Alerta”, é o SC responsável por fazer a consulta dos sensores, que estão disponibilizados em um outro SC do domínio IoT e, em seguida, a análise

dos dados. Deve-se notar que o ator neste caso não é uma pessoa, visto que o “Sistema de Alerta” atua de maneira automática (por exemplo, o sistema está programado para disparar a consulta aos sensores todos os dias em um determinado horário).

A consulta e obtenção de dados dos sensores é realizada por meio da requisição de serviços, de maneira proativa (ver Figura 18 no Capítulo 4). Ao analisar os dados e verificar a presença do mosquito *Aedes*, o “Sistema de Alerta” envia notificações para as entidades responsáveis para a realização de ações de combate e prevenção contra o mosquito.

Figura 22 – Diagrama de caso de uso geral da simulação SoS Sai Dengue



Fonte: Elaborada pelo autor.

Para essa simulação, as entidades responsáveis consideradas foram a Secretaria Municipal de Serviços Públicos e Secretaria Municipal de Saúde da cidade de São Carlos/SP. De acordo com o site da Prefeitura de São Carlos², a Secretaria Municipal de Serviços Públicos tem como responsabilidade a execução de serviços de limpeza, resíduos sólidos e demais serviços de manutenção pública. A Secretaria de Municipal de Saúde é responsável por executar ações de vigilância sanitária e epidemiológica. A Figura 23 ilustra o diagrama de caso de uso de acordo com as entidades que foram consideradas como SCs para a simulação.

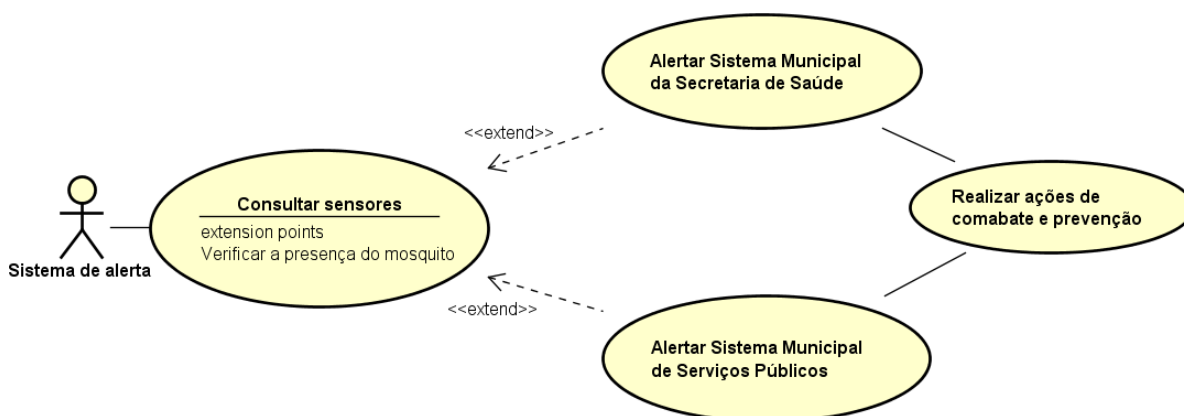
As responsabilidades de cada entidade (SCs) são, respectivamente, as missões individuais. Mesmo não estando participando do SoS, essas entidades tem por responsabilidade a realização de ações de combate, prevenção e limpeza da cidade.

Além do diagrama de caso de uso, um modelo parcial de domínio da simulação também foi elaborado para ilustrar os conceitos iniciais envolvidos na simulação proposta. Esse modelo, ilustrado na Figura 24, indica os SCs envolvidos no SoS, seus atributos iniciais e as ligações entre eles.

Para ilustrar as interações entre os SCs para que o comportamento do SoS possa ser alcançado, foi elaborado o diagrama de sequência (Figura 25), o qual ilustra o fluxo de mensagens trocadas entre os objetos previamente identificados da simulação. Vale ressaltar que, de acordo com a Figura 25, a interação entre os SCs é realizada por meio da invocação de serviços.

² <http://www.saocarlos.sp.gov.br>

Figura 23 – Diagrama de caso de uso da simulação SoS Sai Dengue com as entidades responsáveis



Fonte: Elaborada pelo autor.

5.3.2 Projeto

Nessa fase, foram utilizadas as abstrações definidas para a fase de projeto pelo proponente desta pesquisa de mestrado de acordo com os estudos sobre o desenvolvimento de aplicações IoT e associações com os conceitos de SoS. Cada SC considerado para essa simulação descritos na fase de requisitos foi projetado de forma individual.

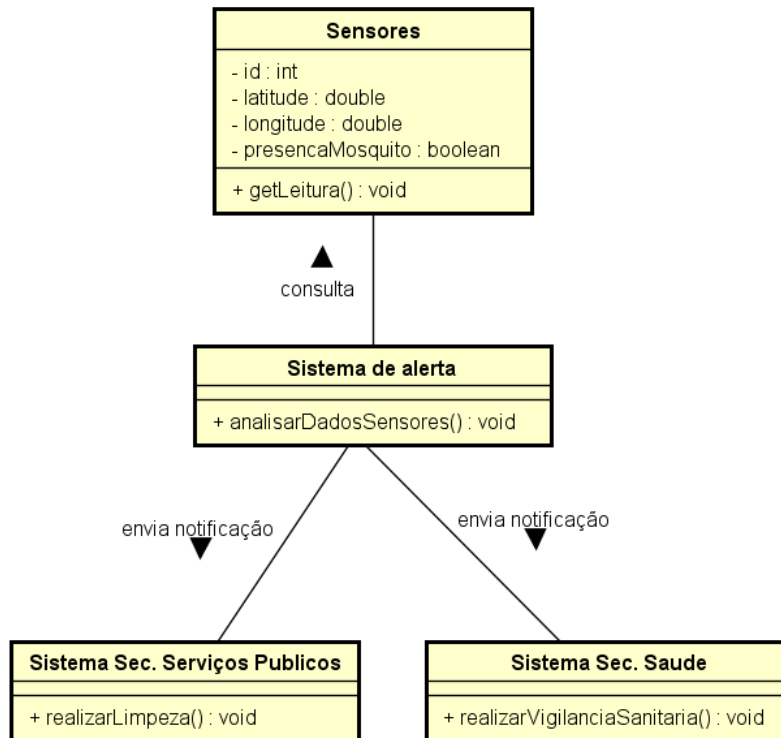
Com os principais sistemas identificados na fase de requisitos, iniciou-se o processo de modelagem dos SCs envolvidos no SoS. Os responsáveis realizaram a modelagem mais detalhada dos sensores, identificando quais os serviços são necessários para que outros SCs pudessem ter acesso aos recursos oferecidos pelos sensores e quais eram os dispositivos envolvidos.

Cada sensor, que faz leitura do ambiente para captar a presença do mosquito, está ligado a um dispositivo, a placa eletrônica, que recebe os sinais capturados da presença do mosquito no ambiente físico e envia para o SC, o sistema gestor de sensores. O recurso, que está disponível para ser acessado por meio da Internet, é acessado pela invocação do serviço “obterDadosSensores”. Na Figura 26 é ilustrado o serviço que permite o acesso aos dados disponibilizados pelos sensores.

Além do recurso obtido pela requisição do serviço “obterDadosSensores”, outro recurso foi adicionado para obter as informações dos sensores. Esse recurso permite obter informações dos sensores, tais como identificador do sensor e posição geográfica (latitude e longitude). Esse recurso fica armazenado no mesmo dispositivo, o sistema gestor de sensores, mudando apenas o tipo de retorno de acordo com a requisição do serviço. Na Figura 27 é ilustrado o serviço “obterInformacoesSensores”.

O Sistema da Secretaria de Serviços Públicos é um sistema de informação que não é dotado de sensor ou atuador. Ele é um sistema que receberá informações de outro sistema

Figura 24 – Modelo conceitual da simulação SoS Sai Dengue



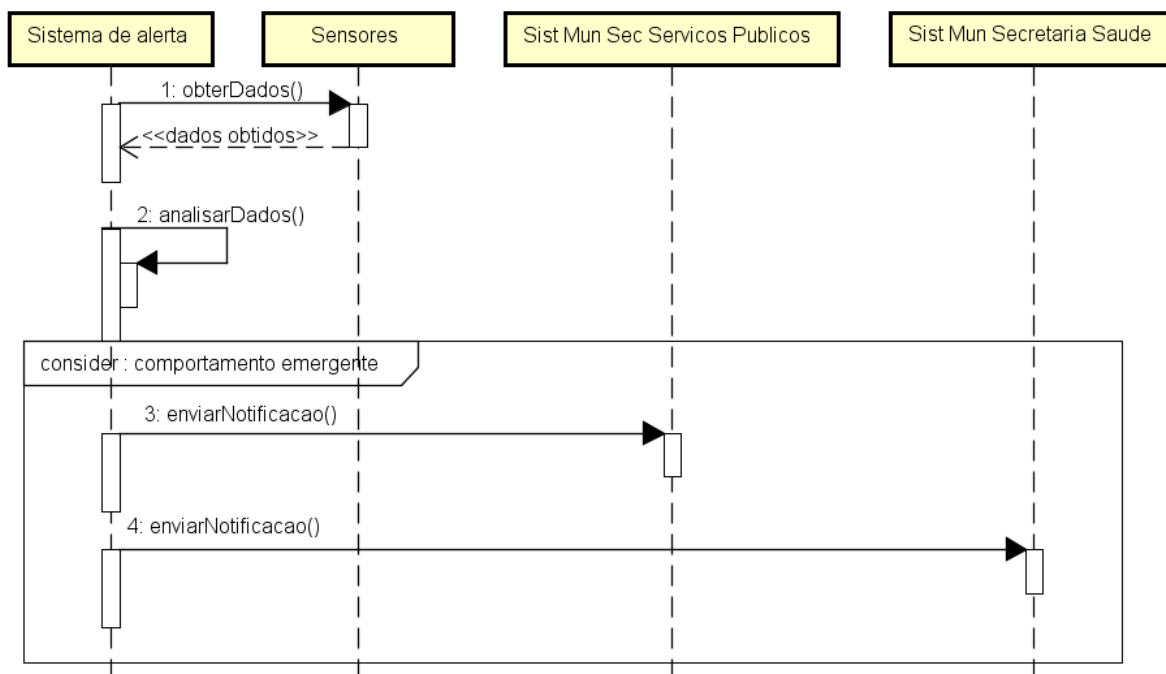
Fonte: Elaborada pelo autor.

contendo dados sobre a presença do mosquito *Aedes*. Esse SC disponibiliza um serviço, denominado “receberAlertaSecSerPublicos”, no qual o recurso acessado pela requisição do serviço está disponível pela Internet. O recurso disponível por esse serviço é de alertar os responsáveis da Secretaria pela realização de ações contra o mosquito. Esse processo é apresentado na Figura 28.

Análogo ao Sistema de Secretaria de Serviços Públicos, o Sistema da Secretaria de Saúde é um sistema de informação que contém o recurso disponível na rede cujo objetivo é alertar as autoridades de saúde sobre a necessidade de obtenção de recursos para tratamento de uma possível epidemia, caso haja a necessidade de acordo com a quantidade de mosquito encontrada pelos sensores. Esse recurso é disponibilizado pela requisição do serviço “receberAlertaSaude” e seu processo é ilustrado na Figura 29.

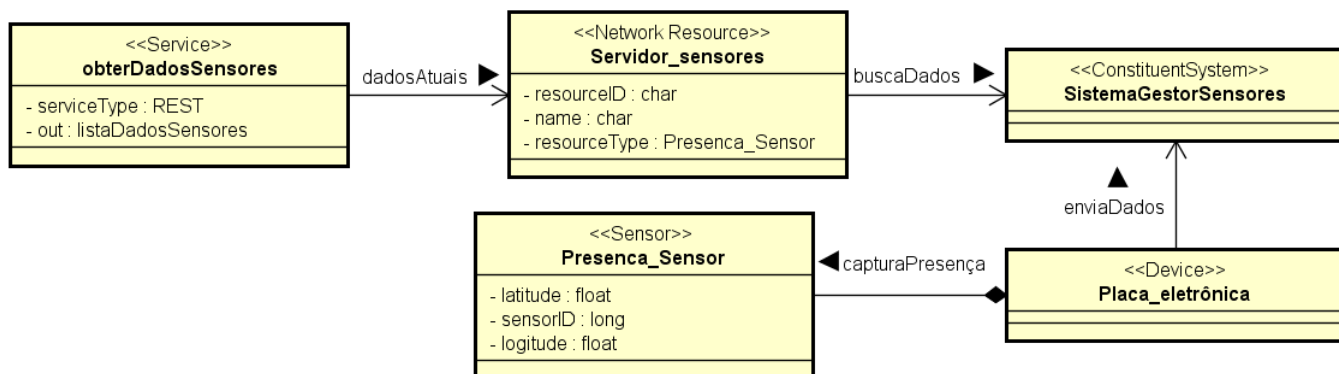
O Sistema de Alerta é um SC que possui o papel de autoridade central e é gerenciado pela Prefeitura. Esse SC é o responsável pelo alcance do comportamento emergente previamente definido. Nessa simulação, o comportamento emergente definido é o “Evitar Epidemia”, no qual o Sistema de Alerta realiza a verificação da existência do mosquito *Aedes* nos locais onde estão localizados os sensores por meio da obtenção dos dados gerados. A Figura 30 ilustra a abstração dos sistemas envolvidos nessa simulação.

Figura 25 – Interação entre os principais SCs definidos da simulação



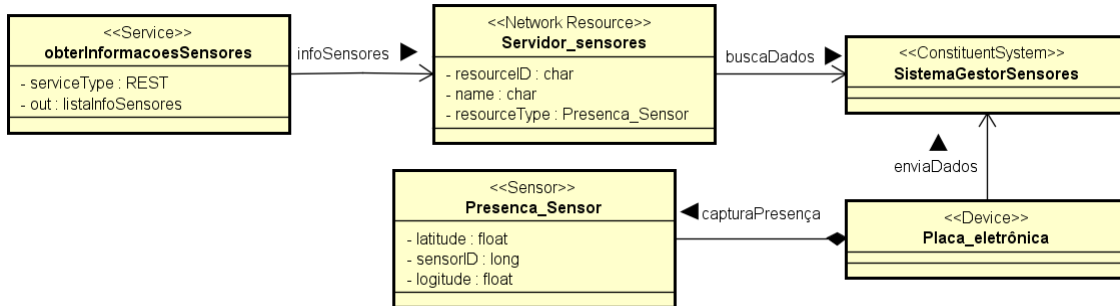
Fonte: Elaborada pelo autor.

Figura 26 – Serviço “ObterDadosSensores”



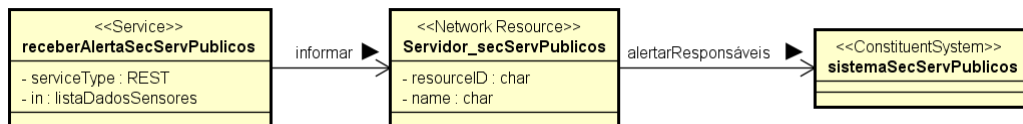
Fonte: Elaborada pelo autor.

Figura 27 – Serviço “obterInformacaoSensores”



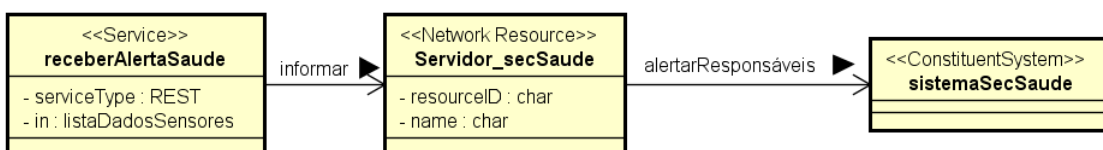
Fonte: Elaborada pelo autor.

Figura 28 – Serviço “receberAlertaSecServPublicos”



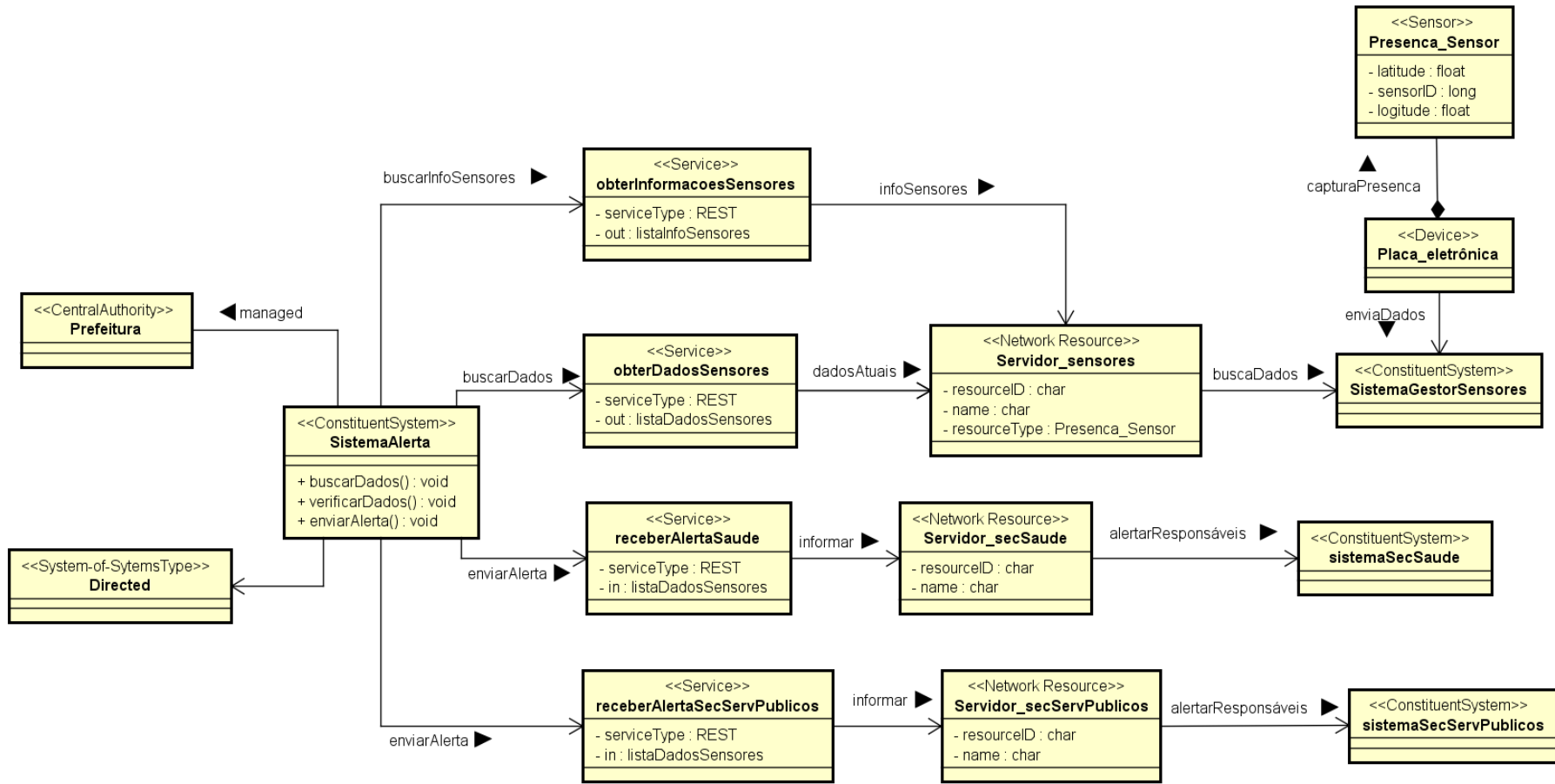
Fonte: Elaborada pelo autor.

Figura 29 – Serviço “receberAlertaSaude”



Fonte: Elaborada pelo autor.

Figura 30 – Sistemas envolvidos na simulação

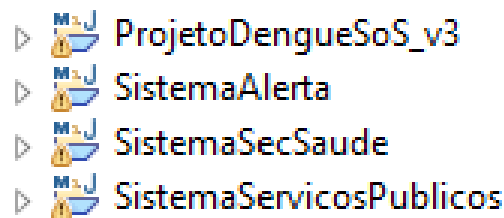


Fonte: Elaborada pelo autor.

5.3.3 Implementação

Com os modelos conceituais elaborados na fase de requisitos e com o detalhamento dos sistemas na fase de projeto por meio das abstrações definidas pelo proponente, deu-se início a fase de implementação. A simulação foi codificada utilizando a linguagem Java, a API JAX-RS para o desenvolvimento de *web services*, e o *framework* hibernate para o mapeamento dos atributos da base de dados, que, neste caso, é utilizado banco de dados relacional MySQL. A simulação foi elaborada em quatro projetos diferentes conforme ilustrado na Figura 31, sendo que cada projeto representa um sistema.

Figura 31 – Projetos da implementação da simulação



Fonte: Elaborada pelo autor.

No projeto **ProjetoDengueSoS_v3** estão implementados os sensores, que geram dados de forma aleatória e enviam para o banco de dados; as entidades mapeadas para o banco de dados; os serviços REST, com os caminhos definidos e os tipos de serviços oferecidos; e o servidor local no qual os serviços estão disponíveis para serem consumidos.

No projeto **SistemaAlerta** está implementado o coordenador central da simulação. O papel do coordenador central é verificar os dados gerados pelos sensores sobre a presença do mosquito *Aedes* nas regiões onde foram implantados os sensores. A obtenção dos dados é feita por meio de requisição de serviço REST.

Nos projetos **SistemaSecSaude** e **SistemaServicosPublicos** estão implementados os serviços de forma independente, ou seja, cada sistema com seu servidor local para o armazenamento e disponibilização dos serviços. Para esta simulação, os serviços implementados para ambos os sistemas foram “receberAlerta” por meio do envio de e-mail. Como não é conhecida a forma como cada secretaria opera computacionalmente, foi considerado que o recebimento de e-mail com o alerta do coordenador central (Sistema de Alerta) seria uma forma das entidades responsáveis tomarem conhecimento da presença do mosquito e realizarem medidas necessárias para o combate e prevenção do mosquito *Aedes*. Os serviços REST implementados pra essa simulação estão descritos na Tabela 15.

Tabela 15 – Serviços REST implementados na simulação

Método HTTP	Caminho	Descrição	Parâmetros
GET	service/obterDadosSensores	Lista de todos os dados produzidos pelos sensores	-
GET	service/obterInformacoesSensores	Lista de todas as informações cadastradas nos sensores	-
GET	service/receberAlertaSaude/bairros	Lista de todos os bairros com a presença do Aedes para a secretaria de saúde	bairros
GET	service/receberAlertaSecServPublicos/bairros	Lista de todos os bairros com a presença do Aedes para a secretaria de serviços públicos	bairros

Fonte: Elaborada pelo autor.

5.4 Funcionamento da simulação

Conforme já descrito anteriormente, a simulação foi desenvolvida utilizando o paradigma de orientação a serviços para que ocorra a integração de SCs presentes em um SoS. Isso foi planejado desde o início para facilitar a integração entre diferentes sistemas de diferentes domínios em um SoS, como, por exemplo, em cidades inteligentes.

Dessa forma, para que a simulação alcance o comportamento emergente por meio da integração entre os SCs, é necessário que todos os servidores onde estão disponíveis os serviços implementados estejam rodando. Como descrito anteriormente, nos projetos **ProjetoDengue-SoS_v3**, **SistemaSecSaude** e **SistemaServicosPublicos** existem servidores em que os serviços estão disponíveis para que o **SistemaAlerta** realize a busca dos dados dos sensores e realize o alerta para as entidades responsáveis caso seja detectado a presença do mosquito *Aedes*. Essa forma de se obter o comportamento emergente é o modelo proativo, como apresentado na Figura 18. O comportamento emergente também poderia ser atingido por meio da interação entre outros sistemas que não fossem o Sistema de Alerta, seguindo o modelo reativo ilustrado na Figura 19.

Seguindo o modelo proativo, os sensores enviam os dados sobre a presença do mosquito em tempos determinados. Os dados são enviados para o sistema gestor de sensores contendo todas as informações de todos os sensores implantados na cidade. O Sistema de Alerta realiza a busca dos dados gerados pelos sensores também em tempos determinados por meio de chamada de serviço. Ao receber os dados, o sistema realiza a verificação da presença do mosquito *Aedes* detectada pelos sensores. Caso seja positiva a presença do mosquito, alertas serão enviados para o Sistema da Secretaria de Serviços Públicos e para o Sistema da Secretaria de Saúde.

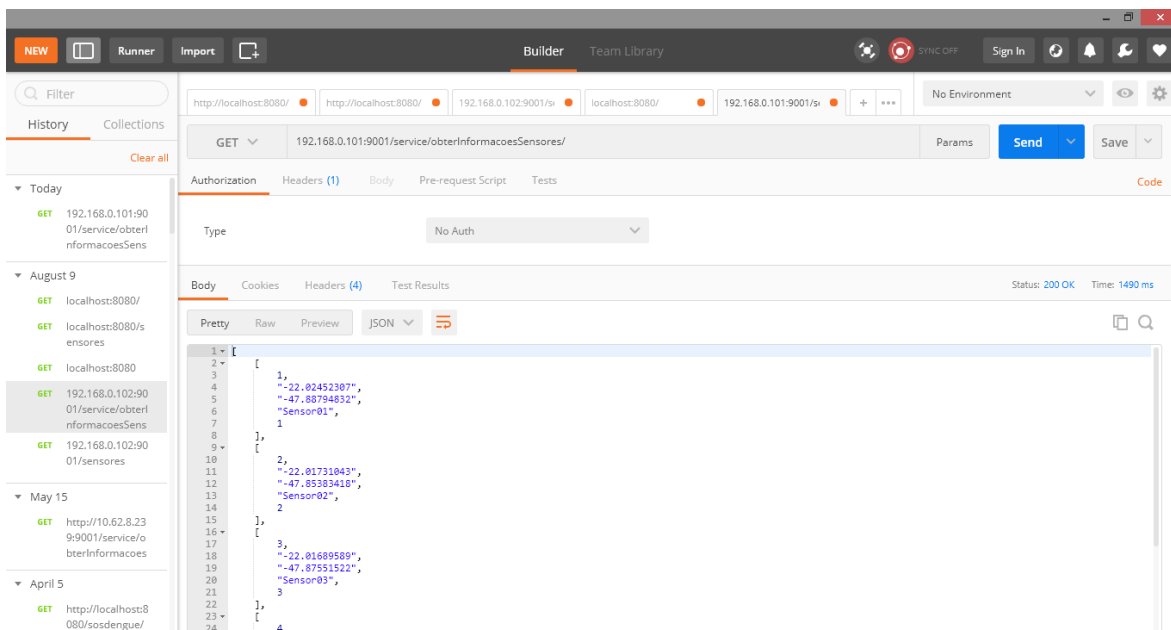
Para testar o funcionamento dos serviços implementados, foi utilizado o Postman³, uma ferramenta que permite facilmente consumir serviços locais e na Internet, enviando dados e efetuando análise sobre as respostas obtidas. Na Figura 32 é ilustrada a interface do Postman

³ <<https://www.getpostman.com/>>

com a requisição do tipo GET do serviço “obterInformacoesSensores”.

Todo o código desenvolvido da simulação está disponível no github⁴ do proponente.

Figura 32 – Consumindo o serviço “obterInformacoesSensores”



Fonte: Elaborada pelo autor.

5.5 Considerações Finais

Como prova de conceito da abordagem proposta, uma simulação foi desenvolvida para verificar a aplicabilidade da abordagem nas diferentes fases do processo de desenvolvimento, sendo elas: requisitos, projeto e implementação. Para o exemplo descrito neste capítulo, a utilização da abordagem se mostrou suficiente para a obtenção de artefatos, principalmente nas fases de requisitos e projeto, contendo as informações necessárias para o desenvolvimento de uma aplicação contendo diversos dispositivos, sendo da IoT ou não, rumo a um SoS.

No Capítulo 6, são apresentadas as conclusões finais deste trabalho, com as contribuições alcançadas, as limitações e trabalhos futuros.

⁴ <github.com/danilloreis01>

CONCLUSÃO

6.1 Visão Geral da Pesquisa e Contribuições

A busca por qualidade e melhoria no desenvolvimento de aplicações em diversos domínios está abrindo novas portas no âmbito de pesquisa em engenharia de software. Um exemplo são as aplicações IoT, que são aplicações compostas de dispositivos heterogêneos e interconectados, presentes em diversos setores, como por exemplo, em cidades inteligentes. A integração de aplicações heterogêneas em ambientes inteligentes pode trazer benefícios maiores e mais bem definidos do que o benefício de apenas uma aplicação, o que leva ao conceito de SoS.

Diante disso, neste trabalho de mestrado procurou-se propor uma solução que apoie o desenvolvimento de aplicações IoT, integrando “coisas” rumo a um SoS. Ambos os campos de pesquisa têm ganhado atenção de setores públicos e privados, inclusive da academia. Portanto, uma abordagem foi apresentada com o objetivo de facilitar o desenvolvimento e a utilização de capacidades disponíveis por meio da integração de diversos sistemas heterogêneos utilizando a arquitetura orientada a serviços.

Essa abordagem apresenta a utilização de conceitos da engenharia de software na fase de requisitos, abstrações dos conceitos de desenvolvimento de aplicações IoT inspiradas no estudo proposto por [Costa, Pires e Delicato \(2016\)](#) e pelo modelo de domínio apresentado por [Bassi et al. \(2013\)](#), além dos conceitos de SoS definidos por [Lane e Epstein \(2013\)](#), [Maier \(1998\)](#) e de uma ontologia proposto por [Abdalla \(2017\)](#) na fase de projeto, e formas de utilização de *web services* na fase de implementação. Como prova de conceito, foi desenvolvida uma simulação de uma aplicação IoT rumo a um SoS do tipo direcionado, um tipo de SoS centralizado no qual o comportamento emergente é definido por meio de um SC denominado de coordenador central.

Dessa forma, a principal contribuição deste trabalho é a abordagem passo a passo para a elaboração de aplicações IoT para se tornar um SoS. A abordagem foi desenvolvida por meio da utilização de conceitos já conhecidos na engenharia de software e também de abstrações de

desenvolvimento de aplicações IoT. Esses conceitos foram unidos com os conceitos e termos de SoS, destacando que a integração de diferentes sistemas de diferentes domínios podem gerar um comportamento maior e mais bem definido. Essa integração entre os SCs (ou coisas) é realizada por meio da utilização da arquitetura orientada a serviços, pois por meio dela é possível que diferentes sistemas possam interoperar facilmente.

A abordagem é útil para desenvolvedores que pretendem construir um sistema que consista na utilização de funcionalidades de outros sistemas, existentes ou não, de forma que um ou mais desses sistemas são do domínio IoT.

6.2 Dificuldades e Limitações

Como esta pesquisa teve como objetivo a construção de uma abordagem para auxiliar o processo de desenvolvimento de aplicações IoT rumo a um SoS, o primeiro passo foi levantar abstrações do domínio IoT em relação ao processo de desenvolvimento. Muitos dos trabalhos encontrados na literatura, principalmente durante a execução do MS, foram relacionados ao desenvolvimento de aplicações pré determinadas para a geração de algum benefício, por exemplo, o desenvolvimento de aplicações IoT para cuidados com a saúde. Esses trabalhos focaram apenas nos benefícios trazidos por meio do uso de sensores e atuadores conectados à Internet, não demonstrando abstrações em relação ao processo de desenvolvimento.

Outro desafio no desenvolvimento deste trabalho de mestrado foi a busca por estudos que relacionassem os termos IoT e SoS com relação ao desenvolvimento de aplicações. Os poucos estudos sobre os assuntos encontram-se em fases iniciais, relatando visões holísticas sobre a união desses termos e também sobre os desafios que o desenvolvimento de SoS a partir de objetos inteligentes presentes no domínio IoT, como em cidades inteligentes, podem trazer, como heterogeneidade e interoperabilidade de dispositivos.

Outra limitação foi a realização da simulação como prova de conceito da abordagem, pois, por ter sido somente uma simulação, não se garante que em um cenário real a abordagem funcionaria da mesma forma. Assim, um estudo de caso completo, considerando um SoS real, envolvendo especialistas das áreas de IoT, SoS e desenvolvedores/programadores seria necessário para validar a abordagem de maneira efetiva.

6.3 Trabalhos Futuros

O foco principal desta pesquisa foi um processo de desenvolvimento de aplicações IoT, de forma que as “coisas” no domínio IoT foram consideradas como SCs e SOA foi utilizada para a integração entre diversos SCs, tanto do domínio IoT quanto fora desse domínio, na formação de um SoS. A partir do desenvolvimento desta pesquisa, outros trabalhos também poderão ser realizados, sendo eles:

- Realizar avaliação com especialistas em IoT e SoS para validar a abordagem, principalmente na fase de análise e projeto, em que abstrações de desenvolvimento de aplicações IoT, e conceitos e termos de SoS são unidos;
- Realizar experimentos controlados com desenvolvedores utilizando a abordagem proposta, a fim de verificar a viabilidade do que foi elaborado;
- Verificar o uso da Linguagem de Modelagem de Sistemas (SysML¹ - Systems Modeling Language) no desenvolvimento de aplicações IoT rumo a um SoS, pois é uma linguagem de modelagem de propósito geral para aplicação na engenharia de sistemas;
- Realizar o estudo de viabilidade da utilização desta abordagem com o estudo de [Patel e Cassou \(2015\)](#), que descreve um conjunto de linguagem de descrição e de arquitetura para o desenvolvimento de aplicações para IoT; e
- Verificar outros tipos de SoS existentes em que aplicações IoT possam fazer parte, por meio de estudos exploratórios e implementação de outras simulações de aplicações, por exemplo, para o tipo de SoS colaborativo, verificando se a abordagem atende a todos os tipos de SoS.

¹ <https://sysml.org/>

REFERÊNCIAS

ABDALLA, G. **Establishment of an Ontology for Systems-of-Systems**. Tese (Doutorado) — Universidade de São Paulo, 2017. Citado nas páginas 71 e 91.

ATZORI, L.; IERA, A.; MORABITO, G. The Internet of Things: A survey. **Computer Networks**, Elsevier, v. 54, n. 15, p. 2787–2805, 2010. Citado nas páginas 30 e 32.

AVI ZINS. **Internet of Bodies: Health IoT**. 2015. Data de acesso: 03-08-2016. Citado na página 33.

AYALA, I.; AMOR, M.; FUENTES, L. Towards a CVL Process to Develop Agents for the IoT. In: SPRINGER. **Proceedings of the 8th International Conference on Ubiquitous Computing and Ambient Intelligence (UCAmI 2014)**. Belfast, Irlanda do Norte, 2014. p. 304–311. Citado na página 60.

AZIZ, M. W.; SHEIKH, A. A.; FELEMBAN, E. A. Requirement Engineering Technique for Smart Spaces. In: **Proceedings of the International Conference on Internet of Things and Cloud Computing (ICC 2016)**. Nova York, EUA: ACM, 2016. p. 1–7. Citado nas páginas 24, 25 e 57.

BANDYOPADHYAY, D.; SEN, J. Internet of Things: Applications and Challenges in Technology and Standardization. **Wireless Personal Communications**, v. 58, n. 1, p. 49–69, 2011. Citado na página 32.

BARI, N.; MANI, G.; BERKOVICH, S. Internet of Things as a Methodological Concept. In: **Proceedings of the 4th International Conference on Computing for Geospatial Research and Application (COM.Geo 2013)**. São José, EUA: IEEE, 2013. p. 48–55. Citado na página 30.

BASSI, A.; BAUER, M.; FIEDLER, M.; KRANENBURG, R. **Enabling things to talk**. 1. ed. Berlin, Alemanha: Springer-Verlag Berlin Heidelberg, 2013. Citado nas páginas 48, 61, 71 e 91.

BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. Developing multi-agent systems with a fipa-compliant agent framework. **Software: Practice and Experience**, Wiley Online Library, v. 31, n. 2, p. 103–128, 2001. Citado nas páginas 60 e 62.

BELLY, B. **Bigbelly smart solutions for cities**. 2003. Acessado em: 25-07-2016. Disponível em: <<https://bigbelly.com/solutions/city/>>. Citado na página 35.

BOUZIAT, T.; CAMPS, V.; COMBETTES, S. A cooperative SoS architecting approach based on adaptive multi-agent systems. In: ACM. **Proceedings of the 6th International Workshop on Software Engineering for Systems-of-Systems (SESoS 2018)**. Gotemburgo, Suécia, 2018. p. 8–16. Citado na página 36.

CAVALCANTE, E.; CACHO, N.; LOPES, F.; BATISTA, T.; OQUENDO, F. Thinking smart cities as systems-of-systems: A perspective study. In: ACM. **Proceedings of the 2nd International Workshop on Smart (SmartCitie 2016)**. Trento, Itália, 2016. p. 9–12. Citado nas páginas 26 e 76.

CHEN, X.; LI, A.; ZENG, X.; GUO, W.; HUANG, G. Runtime Model Based Approach to IoT Application Development. **Frontiers of Computer Science**, Springer-Verlag New York, Secaucus, EUA, v. 9, n. 4, p. 540–553, 2015. Citado na página 24.

COETZEE, L.; EKSTEEN, J. The Internet of Things - promise for the future? An introduction. In: **Proceedings of the IST-Africa Conference Proceedings (IST 2011)**. Gaborone, Botswana: IEEE, 2011. p. 1–9. Citado nas páginas 23, 29 e 71.

COMER, D. E. **Redes de Computadores e Internet**. 6. ed. Porto Alegre, Brasil: Bookman Editora, 2016. Citado na página 23.

COSTA, B.; PIRES, P. F.; DELICATO, F. C. Modeling IoT Applications with SysML4IoT. In: IEEE. **Proceedings of the 42th Conference on Euromicro Software Engineering and Advanced Applications (SEAA 2016)**. Limassol, Chipre, 2016. p. 157–164. Citado nas páginas 23, 24, 26, 47, 48, 49, 59, 61, 67, 71, 74, 75 e 91.

DAHMANN, J. The state of systems of systems engineering knowledge sources. In: **Proceedings of the 10th System of Systems Engineering Conference (SoSE 2015)**. Santo Antônio, EUA: IEEE, 2015. p. 475–479. Citado na página 38.

DE, S.; BARNAGHI, P.; BAUER, M.; MEISSNER, S. Service modelling for the Internet of Things. In: IEEE. **Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS 2011)**. Estetino, Polônia, 2011. p. 949–955. Citado na página 61.

DERHAMY, H.; ELIASSON, J.; DELSING, J.; PRILLER, P. A survey of commercial frameworks for the Internet of Things. In: **Proceedings of the 20th IEEE Conference on Emerging Technologies Factory Automation (ETFA 2015)**. Luxemburgo: IEEE, 2015. p. 1–8. Citado na página 30.

ERL, T. **Service-oriented architecture: a field guide to integrating XML and Web services/Thomas Erl**. Upper Saddle River, EUA: Prentice Hall, 2004. 536 p. Citado nas páginas 43 e 44.

EVANS, D. **A Internet das Coisas: Como a próxima evolução da internet está mudando tudo**. 2011. Acessado em: 25-07-2016. Disponível em: <http://www.cisco.com/c/dam/global/pt_br/assets/executives/pdf/internet_of_things_iiot_ibsg_0411final.pdf>. Citado na página 30.

FORTINO, G.; GARRO, A.; MASCILLARO, S.; RUSSO, W. Using event-driven lightweight DSC-based agents for MAS modelling. **International Journal of Agent-Oriented Software Engineering**, Inderscience Publishers, v. 4, n. 2, p. 113–140, 2010. Citado nas páginas 60 e 62.

FORTINO, G.; GUERRIERI, A.; RUSSO, W.; SAVAGLIO, C. Towards a development methodology for smart object-oriented iot systems: A metamodel approach. In: **Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC 2015)**. Hong Kong, China: IEEE, 2015. p. 1297–1302. Citado nas páginas 60 e 62.

FORTINO, G.; LACKOVIC, M.; RUSSO, W.; TRUNFIO, P. A Discovery Service for Smart Objects over an Agent-Based Middleware. In: SPRINGER. **Proceedings of the 11th International Conference on Internet and Distributed Computing Systems (IDCS 2013)**. Hanzou, China, 2013. p. 281–293. Citado nas páginas 60 e 62.

FURTADO, C.; PEREIRA, V.; AZEVEDO, L.; BAIÃO, F.; SANTORO, F.; MAGDALENO, A.; CAPELLI, C.; NUNES, V. **Arquitetura Orientada a Serviço-Conceituação**. Rio de Janeiro, Brasil, 2009. Citado nas páginas 26, 41 e 42.

HAUGEN, Ø.; MØLLER-PEDERSEN, B.; OLDEVIK, J.; OLSEN, G. K.; SVENDSEN, A. Adding Standardized Variability to Domain Specific Languages. In: IEEE. **Proceedings of the 12th International Software Product Line Conference (SPLC 2008)**. Limerick, Irlanda, 2008. p. 139–148. Citado na página 60.

ISO/IEC-24765. Systems and software engineering – vocabulary. **Technical Report, International Organization for Standardization and International Electrotechnical Commission**, 2010. Citado na página 36.

JIA, X.; FENG, Q.; FAN, T.; LEI, Q. RFID technology and its applications in Internet of Things (IoT). In: **Proceedings of the 2nd International Conference on Consumer Electronics, Communications and Networks (CECNet 2012)**. Yichang, China: IEEE, 2012. p. 1282–1285. Citado na página 31.

KITCHENHAM, B.; CHARTERS, S. **Guidelines for Performing Systematic Literature Reviews in Software Engineering**. [S.l.], 2007. Citado nas páginas 51, 52, 53 e 54.

KRANENBURG, R. V.; ANZELMO, E.; BASSI, A.; CAPRIO, D.; DODSON, S.; RATTO, M. **The internet of things**. 2011. Citado na página 31.

LANE, J. A.; EPSTEIN, D. **What is a System of Systems and Why Should I Care?** USC - University of Southern California, 2013. 14 p. Disponível em: <<http://csse.usc.edu/TECHRPTS/2013/reports/usc-csse-2013-500.pdf>>. Citado nas páginas 37, 38, 39, 40 e 91.

LARMAN, C. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objeto e ao desenvolvimento iterativo**. 3^a. ed. [S.l.]: Bookman, 2007. ISBN 978-85-60031-52-8. Citado nas páginas 63 e 68.

LASKEY, K.; ESTEFAN, J. A.; MCCABE, F. G.; THORNTON, D. Reference architecture foundation for service oriented architecture version 1.0. **Oasis, Committee Draft**, v. 2, 2009. Citado nas páginas 26, 40 e 41.

LEWIS, G.; MORRIS, E.; SIMANTA, S.; SMITH, D. Service orientation and systems of systems. **IEEE software**, IEEE, n. 1, p. 58–63, 2011. Citado nas páginas 26, 67, 74 e 75.

MAIA, P.; CAVALCANTE, E.; GOMES, P.; BATISTA, T.; DELICATO, F. C.; PIRES, P. F. On the Development of Systems-of-Systems Based on the Internet of Things: A Systematic Mapping. In: **The proceedings of European Conference on Software Architecture Workshops (ECSAW 2014)**. Nova York, EUA: ACM, 2014. p. 1–8. Citado nas páginas 25, 26 e 60.

MAIER, M. W. Architecting principles for systems-of-systems. **Systems Engineering**, John Wiley & Sons, Inc., v. 6, n. 1, p. 267–284, 1998. Citado nas páginas 36, 38 e 91.

MIRKO, P. (Ed.). **Inspirando a Internet das Coisas**. 1. ed. Alexandra Institute, 2012. Disponível em: <https://iotcomicbook.files.wordpress.com/2013/10/iot_comic_book_special_br.pdf>. Citado nas páginas 32, 33, 35 e 36.

MULANI, T. T.; PINGLE, S. V. Internet of things. **International Research Journal of Multi-disciplinary Studies**, v. 2, n. 3, 2016. Citado na página 30.

- NASA. **Rover**. 2011. Acessado em: 25-07-2016. Disponível em: <<http://mars.jpl.nasa.gov/msl/mission/rover/>>. Citado na página 40.
- NIELSEN, C. B.; LARSEN, P. G.; FITZGERALD, J.; WOODCOCK, J.; PELESKA, J. Systems of systems engineering: Basic concepts, model-based techniques, and research directions. **ACM Computing Surveys**, ACM, New York, USA, v. 48, n. 2, p. 1–41, 2015. Citado nas páginas 36, 37 e 39.
- OLIVEIRA, L. B. R. de; FELIZARDO, K. R.; FEITOSA, D.; NAKAGAWA, E. Y. Reference models and reference architectures based on service-oriented architecture: a systematic review. In: SPRINGER. **European Conference on Software Architecture**. [S.l.], 2010. p. 360–367. Citado na página 40.
- PACIFICO, A. L. A internet das coisas vai mudar – e muito – a nossa vida. Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD), jun. 2016. Disponível em: <<https://www.cpqd.com.br/insight/internet-das-coisas-vai-mudar-e-muito-nossa-vida/>>. Citado na página 24.
- PATEL, P.; CASSOU, D. Enabling high-level application development for the internet of things. **Journal of Systems and Software**, Elsevier, v. 103, p. 62–84, 2015. Citado nas páginas 24, 44, 46, 59, 61 e 93.
- PAUTASSO, C.; ZIMMERMANN, O.; LEYMANN, F. Restful web services vs. big’web services: making the right architectural decision. In: ACM. **Proceedings of the 17th international conference on World Wide Web**. [S.l.], 2008. p. 805–814. Citado na página 42.
- PETERSEN, K.; FELDT, R.; MUJTABA, S.; MATTSSON, M. Systematic mapping studies in software engineering. In: **The proceedings of 12th International conference on evaluation and assessment in software engineering (EASE 2008)**. Swinton, UK: British Computer Society, 2008. v. 17, n. 1. Citado na página 51.
- PETERSEN, K.; VAKKALANKA, S.; KUZNIARZ, L. Guidelines for conducting systematic mapping studies in software engineering: An update. **Information and Software Technology**, v. 64, n. , p. 1 – 18, 2015. ISSN 0950-5849. . Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0950584915000646>>. Citado na página 54.
- PORTAL DA MEDICINA. **Internet das coisas: entenda os seus impactos no mundo da medicina**. 2016. Data de acesso: 3 ago. 2016. Disponível em: <<http://portaltelemedicina.com.br/internet-das-coisas-entenda-os-seus-impactos-no-mundo-da-medicina/>>. Citado na página 33.
- PRAMUDIANTO, F.; INDRA, I. R.; JARKE, M. Model driven development for internet of things application prototyping. In: **SEKE**. [S.l.: s.n.], 2013. p. 703–708. Citado nas páginas 59 e 61.
- PREHOFER, C. Models at rest or modelling restful interfaces for the internet of things. In: **2nd World Forum on Internet of Things (WF-IoT 2015)**. Milão, Itália: IEEE, 2015. p. 251–255. Citado na página 61.
- PRESSMAN, R.; MAXIM, B. **Engenharia de Software-8ª Edição**. [S.l.]: McGraw Hill Brasil, 2016. Citado na página 63.
- ROTONDI, R. M. A. B. D. **Towards a definition of the Internet of Things (IoT)**. IEEE Internet of Things, 2015. Acessado em: 01-08-2016. Disponível em: <http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf>. Citado na página 31.

SEO, D.; SHIN, D.; BAEK, Y.-M.; SONG, J.; YUN, W.; KIM, J.; JEE, E.; BAE, D.-H. Modeling and verification for different types of system of systems using prism. In: **4th International Workshop on Software Engineering for Systems-of-Systems (SESOS 2016)**. New York, USA: ACM, 2016. p. 12–18. Citado na página 25.

SILVA, D. F.; SOUZA, V. M.; ELLIS, D. P.; KEOGH, E. J.; BATISTA, G. E. Exploring low cost laser sensors to identify flying insect species. **Journal of Intelligent & Robotic Systems**, Springer, v. 80, n. 1, p. 313–330, 2015. Citado na página 80.

SINHA, R.; NARULA, A.; GRUNDY, J. Parametric statecharts: designing flexible iot apps: deploying android m-health apps in dynamic smart-homes. In: ACM. **Proceedings of the Australasian Computer Science Week Multiconference**. [S.l.], 2017. p. 28. Citado nas páginas 60 e 61.

SOMMERVILLE, I. **Engenharia de software. Tradução Ivan Bosnic e Kalinka G. de O. Gonçalves; revisão técnica Kechi Hiramã**-. [S.l.]: São Paulo: Pearson Prentice Hall, 2011. Citado na página 63.

SOUZA, V. A. S. M. d. *et al.* Uma arquitetura orientada a serviços para desenvolvimento, gerenciamento e instalação de serviços de rede. [sn], 2006. Citado nas páginas 41 e 42.

SURESH, P.; DANIEL, J. V.; PARTHASARATHY, V.; ASWATHY, R. H. A state of the art review on the Internet of Things (IoT) history, technology and fields of deployment. In: **6th International Conference on Science Engineering and Management Research (ICSEMR 2014)**. Chennai, India: IEEE, 2014. p. 1–8. Citado na página 31.

TAN, L.; WANG, N. Future internet: The internet of things. In: **3rd International Conference on Advanced Computer Theory and Engineering (ICACTE 2010)**. Chengdu, China: IEEE, 2010. p. 376–380. ISSN 2154-7491. Citado nas páginas 23 e 30.

VARGAS, I. G.; GOTTARDI, T.; BRAGA, R. T. V. Approaches for integration in system of systems: a systematic review. In: IEEE. **Software Engineering for Systems-of-Systems (SESoS), 2016 IEEE/ACM 4th International Workshop on**. [S.l.], 2016. p. 32–38. Citado nas páginas 26 e 49.

WAGH, K.; THOOL, R. A comparative study of soap vs rest web services provisioning techniques for mobile host. **Journal of Information Engineering and Applications**, v. 2, n. 5, p. 12–16, 2012. Citado na página 43.

XIE, K.; CHEN, H.; CUI, L. Pmda: A physical model driven software architecture for internet of things. **Computer Research and Development**, v. 50, n. 6, p. 1185–1197, 2013. Citado na página 59.

XIE, K.; CHEN, H.; HUANG, X.; CUI, L. Low cost iot software development-ingredient transformation and interconnection. In: IEEE. **Parallel and Distributed Systems (ICPADS), 2015 IEEE 21st International Conference on**. [S.l.], 2015. p. 44–51. Citado nas páginas 59 e 61.

ZAMBONELLI, F. Key abstractions for iot-oriented software engineering. **IEEE Software**, IEEE, v. 34, n. 1, p. 38–45, 2017. Citado na página 61.

