

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
DEPARTAMENTO DE COMPUTAÇÃO**

Marina Rocha Daros

**DESENVOLVIMENTO DE UMA ARQUITETURA
COMPUTACIONAL PARA SENSORIAMENTO
PARTICIPATIVO EM *SMART CITIES***

Araranguá

2018

Marina Rocha Daros

**DESENVOLVIMENTO DE UMA ARQUITETURA
COMPUTACIONAL PARA SENSORIAMENTO
PARTICIPATIVO EM *SMART CITIES***

Trabalho de Conclusão de Curso submetido à Universidade Federal de Santa Catarina para a obtenção do Grau de Bacharel em Engenharia de Computação.

Universidade Federal de Santa Catarina: Prof. Anderson Luiz Fernandes Perez, Dr.

Araranguá

2018

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Daros, Marina Rocha

Desenvolvimento de uma arquitetura computacional
para sensoriamento participativo em smart cities /
Marina Rocha Daros ; orientador, Anderson Luiz
Fernandes Perez, 2018.

126 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Campus
Araranguá, Graduação em Engenharia de Computação,
Araranguá, 2018.

Inclui referências.

1. Engenharia de Computação. 2. Smart City. 3.
Sensoriamento Participativo. 4. Arquitetura
Computacional. 5. Aplicativo Android. I. Perez,
Anderson Luiz Fernandes. II. Universidade Federal
de Santa Catarina. Graduação em Engenharia de
Computação. III. Título.

Marina Rocha Daros

**DESENVOLVIMENTO DE UMA ARQUITETURA
COMPUTACIONAL PARA SENSORIAMENTO
PARTICIPATIVO EM *SMART CITIES***

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Engenharia de Computação”, e aprovado em sua forma final pela Universidade Federal de Santa Catarina.

Araranguá, 2 de julho de 2018.



Prof^ª. Eliane Pozzebon, Dr^ª.
Coordenadora de Curso

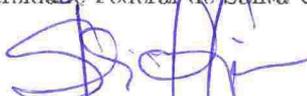
Banca Examinadora:



Prof. Anderson Luiz Fernandes Perez, Dr.
Universidade Federal de Santa Catarina



Prof. Gustavo Mello Machado, Dr.
Universidade Federal de Santa Catarina



Prof^ª. Analúcia Schiaffino Morales, Dr^ª.
Universidade Federal de Santa Catarina

Dedico este trabalho à minha família, principalmente meus pais, e ao meu orientador, Anderson Luiz Fernandes Perez.

AGRADECIMENTOS

Agradeço a meus pais por sempre terem me proporcionado meios para ter acesso à educação e por me motivarem a concluir este trabalho.

Agradeço ao meu irmão, Renan Rocha Darós, por ter me ajudado a entender quando eu não entendia, por ter me guiado em trabalhos que eu não sabia por onde começar e pela companhia durante todos esses anos no trajeto de ida e volta.

Agradeço ao meu orientador, Anderson Luiz Fernandes Perez, cuja instrução foi imprescindível para este trabalho. Sem suas diretrizes teria sido bem mais difícil alcançar os objetivos estabelecidos.

Queria também agradecer formalmente a equipe do projeto InterSCity da Universidade de São Paulo, especialmente Arthur Del Esposte e Dylan Guedes, por ter me ajudado na configuração da plataforma.

Agradeço a todos os amigos e colegas que fiz neste período da graduação por terem dado apoio e compartilhado a apreensão diante dos desafios do curso.

E agradeço, por fim, ao Igor Venicius de Sousa pela companhia, apoio e por ter acreditado na minha capacidade em todas as vezes que duvidei dela.

Nothing in life is to be feared, it is only to be understood. Now is the time to understand more, so that we may fear less.

Marie Curie

RESUMO

As cidades são lugares importantes, pois são cruciais para o desenvolvimento sócio-econômico de uma região. Por isso, a população urbana vem crescendo nos últimos anos. Este aumento da densidade populacional nos grandes centros e o envelhecimento populacional trouxe a necessidade de alocar recursos, como energia e transporte público, de forma mais eficiente. O termo *Smart City* refere-se ao uso das Tecnologias da Informação e Comunicação (TIC) com este intuito. Através do paradigma de Sensoriamento Participativo, que alia as TICs e os usuários de *smartphones*, é possível entender características da dinâmica de uma cidade e as urgências dos cidadãos que vivem nela. Neste trabalho é proposta uma arquitetura computacional para Sensoriamento Participativo em *Smart Cities*, tornando possível que os participantes enviem registros de ocorrências acerca da cidade, através de um aplicativo Android, para um *middleware*. Tais problemas podem ser relacionados à segurança, infraestrutura e mobilidade da cidade. A partir dos dados enviados, utiliza-se *Self-Organizing Map* (SOM) para inferir informações que servem para aprimorar o entendimento do que acontece na cidade, também com o objetivo de facilitar a decisão dos representantes políticos, analisando assuntos considerados mais críticos.

Palavras-chave: *Smart City*, Sensoriamento Participativo, Arquitetura Computacional, Aplicativo Android, *Self-Organizing Map*.

ABSTRACT

Cities are important places because they are substantial to the social and economical development of a region. Therefore, urban population has been growing in the last years. This growth of population density in downtowns and population ageing have led to a necessity of resource distribution, such as power and public transportation, in a more effective way. The term Smart City refers to the employment of Information and Communication Technology (ICT) with this goal. Using, Participatory Sensing, a paradigm which consists in using ICT and smartphones users to act as sensors, it is possible to understand the dynamic of a city and the most urgent necessities of its citizen. In this thesis, a Participatory Sensing computer architecture to Smart Cities is proposed, enabling its users to send records of events about the city to a middleware with an Android app. Such events can be related to the city safety, infrastructure and mobility. After running the Self-Organizing Map technique over the sent data, intelligent information about the city has been known. Therefore, it is possible to understand the characteristics of city's regions and help officials in decision making.

Keywords: Smart City, Participatory Sensing, Computer Architecture, Android app, Self-Organizing Map.

LISTA DE FIGURAS

Figura 1	Componentes característicos de uma SC.	36
Figura 2	Tipos de aplicações de SC.	37
Figura 3	Arquitetura da Plataforma SOFIA2.	43
Figura 4	Arquitetura da plataforma InterSCity.	45
Figura 5	Funcionamento de uma RSP.	48
Figura 6	Geração de informação contextual em uma RSP.	51
Figura 7	Arquitetura proposta.	57
Figura 8	Tipos de ocorrências.	58
Figura 9	Diagrama de Caso de Uso.	59
Figura 10	Troca de informações no acesso.	61
Figura 11	Principais processos entre a aplicação e o <i>middleware</i>	61
Figura 12	Mapa Auto-Organizável.	63
Figura 13	Dados de entrada e saída do SOM.	64
Figura 14	Interfaces do aplicativo InterageAru.	67
Figura 15	Fluxo de telas.	68
Figura 16	Cadastro de usuário.	71
Figura 17	Usuário cadastrado na plataforma.	71
Figura 18	LOG exibe <code>uuid</code> atribuído.	72
Figura 19	Entrar com CPF já cadastrado.	72
Figura 20	Informações registradas na plataforma.	72
Figura 21	LOG exibe <code>uuid</code> após busca.	72
Figura 22	Mensagem informa que não está cadastrado.	73
Figura 23	LOG com <code>uuid</code> não retornado.	73
Figura 24	Envio de ocorrência.	74
Figura 25	Ocorrência cadastrada na plataforma.	74
Figura 26	LOG exibe <code>uuid</code> e <code>url</code>	74
Figura 27	Simulação de cadastro de recurso com Postman.	76
Figura 28	Retorno da simulação de cadastro de recurso.	76
Figura 29	Tempo de resposta de cadastro de recurso com o aplicativo.	77
Figura 30	Simulação de envio de ocorrência com Postman.	78
Figura 31	Retorno de envio de ocorrência com Postman.	78

Figura 32 Tempo de resposta de envio de ocorrência com Interage-
Aru. 79

Figura 33 Busca de recurso com Postman. 79

Figura 34 Tempo de resposta com o aplicativo. 80

Figura 35 Mapa de componentes do primeiro teste de ocorrência.. 84

Figura 36 *Clusters* do primeiro teste de ocorrência. 85

Figura 37 Mapa de componentes do segundo teste de ocorrência. . 87

Figura 38 *U-matrix* do segundo teste de ocorrência. 87

Figura 39 *Clusters* do segundo teste de ocorrência. 88

Figura 40 Mapa de componentes do teste de aleatoriedade. 90

Figura 41 *U-matrix* do teste de aleatoriedade. 90

Figura 42 *Clusters* do teste de aleatoriedade. 91

Figura 43 Microservices communication on resource creation. 120

Figura 44 Microservices communication on data stream. 121

Figura 45 Design roadmap. 122

Figura 46 Architecture. 124

LISTA DE TABELAS

Tabela 1	Conceitos relacionados à SC.....	33
Tabela 2	Principais recursos dos GEs.....	44
Tabela 3	Porta dos microsserviços.....	66
Tabela 4	Método de solicitação por funcionalidade.....	69
Tabela 5	Atributos do primeiro conjunto de dados.....	82
Tabela 6	Índices por natureza e ocorrência.....	83
Tabela 7	Latitudes e Longitudes.....	92
Tabela 8	Taxa de sucesso de envio.....	94

LISTA DE ABREVIATURAS E SIGLAS

SC	<i>Smart City</i>	25
TIC	Tecnologias da Informação e Comunicação	25
SP	Sensoriamento Participativo	26
RSP	Rede de Sensoriamento Participativo	26
RSSF	Redes de Sensores Sem Fio	26
IBGE	Instituto Brasileiro de Geografia e Estatística	27
IoT	<i>Internet of Things</i>	33
HCI	<i>Human-Computer Interaction</i>	39
GPS	<i>Global Positioning System</i>	41
SOFIA	<i>Smart Objects for Intelligent Applications</i>	43
TCP	<i>Transmission Control Protocol</i>	43
MQTT	<i>Message Queuing Telemetry Transport</i>	43
HTTP	<i>Hypertext Transfer Protocol</i>	43
API	<i>Application Programming Interface</i>	44
GE	<i>Generic Enabler</i>	44
USP	Universidade de São Paulo	45
XML	<i>eXtensible Markup Language</i>	54
PEIR	<i>Personal Environmental Impact Report</i>	54
JSON	<i>JavaScript Object Notation</i>	54
PHP	<i>PHP: Hypertext Preprocessor</i>	55
JSON	<i>JavaScript Object Notation</i>	60
SOM	<i>Self-Organizing Map</i>	62
RNA	Rede Neural Artificial	62
BMU	<i>Best Matching Unit</i>	63
IDE	<i>Integrated Development Environment</i>	65
JDK	<i>Java Development Kit</i>	65

LISTA DE CÓDIGOS

Código 1	Containers do microsserviço <code>Data Collector</code>	65
Código 2	Modelo JSON de cadastro de uma nova <code>capability</code> . .	70
Código 3	Modelo JSON de cadastro de um recurso	70
Código 4	Modelo JSON de registro de ocorrência	70
Código 5	Lista de clusters do primeiro teste de ocorrência	85
Código 6	Lista de clusters do segundo teste de ocorrência	88
Código 7	Lista de clusters do teste de aleatoriedade	91
Código 8	Relatório de ocorrências mais frequentes	93
Código 9	Salva os dados em formato <code>.csv</code>	109
Código 10	Processa dados sobre <code>.csv</code>	110
Código 11	Relatório sobre dados do <code>.csv</code>	111
Código 12	Gerador de recursos e dados aleatórios	112

SUMÁRIO

1 INTRODUÇÃO	25
1.1 JUSTIFICATIVA E MOTIVAÇÃO	26
1.2 OBJETIVOS	27
1.2.1 Geral	27
1.2.2 Específicos	27
1.3 METODOLOGIA	28
1.4 ORGANIZAÇÃO DO TRABALHO	28
2 FUNDAMENTOS DE <i>SMART CITY</i>	31
2.1 DEFINIÇÃO DE <i>SMART CITIES</i>	31
2.2 CARACTERÍSTICAS DE UMA <i>SMART CITY</i>	34
2.3 APLICAÇÕES	36
2.3.1 Economia Inteligente	37
2.3.2 Mobilidade Inteligente	38
2.3.3 Meio Ambiente Inteligente	38
2.3.4 Pessoas Inteligentes	39
2.3.5 Vida Inteligente	39
2.3.6 Governo Inteligente	40
2.4 TECNOLOGIA	40
2.4.1 <i>Middleware</i>	42
2.4.2 SOFIA	43
2.4.3 FIWARE	44
2.4.4 InterSCity	45
3 SENSORIAMENTO PARTICIPATIVO	47
3.1 SENSORIAMENTO DISTRIBUÍDO	47
3.2 DEFINIÇÃO DE SENSORIAMENTO PARTICIPATIVO ..	48
3.3 CARACTERÍSTICAS	50
3.3.1 Desafios	52
3.3.2 Oportunidades	53
3.4 INICIATIVAS DE SENSORIAMENTO PARTICIPATIVO ..	54
4 ARQUITETURA COMPUTACIONAL PARA SEN- SORIAMENTO PARTICIPATIVO	57
4.1 ARQUITETURA LÓGICA	57
4.1.1 <i>Middleware</i>	60
4.2 COMUNICAÇÃO	60
4.2.1 Registro	60
4.3 PROCESSAMENTO DE DADOS	62
4.3.1 Mapa Auto-Organizável	62

4.3.2 Acesso à Informação	64
5 APLICAÇÃO DESENVOLVIDA	65
5.1 IMPLEMENTAÇÃO	65
5.2 CONFIGURAÇÃO DO INTERSCITY	65
5.3 INTERAGEARU	66
5.3.1 Activities	67
5.4 COMUNICAÇÃO COM INTERSCITY	69
5.5 APLICATIVO EM FUNCIONAMENTO	71
6 AVALIAÇÃO DO SISTEMA PROPOSTO	75
6.1 PROCEDIMENTOS ADOTADOS	75
6.2 AVALIAÇÃO DO APLICATIVO	75
6.3 AVALIAÇÃO DA FASE DE PROCESSAMENTO DOS DA- DOS	80
6.3.1 Geração de Dados	81
6.3.2 Tratamento dos Dados	82
6.3.3 Teste de ocorrência	83
6.3.4 Teste de Aleatoriedade	89
6.3.5 Relatórios	92
6.4 AVALIAÇÃO DA PLATAFORMA	93
6.5 DIFICULDADES	94
6.5.1 Middleware	94
6.5.2 Desenvolvimento Android	95
6.5.3 Processamento de Dados	95
7 CONSIDERAÇÕES FINAIS	97
7.1 TRABALHOS FUTUROS	98
REFERÊNCIAS	99
APÊNDICE A – Algoritmos utilizados no processamento de dados	109
ANEXO A – Documentação do InterSCity	119

1 INTRODUÇÃO

Nas últimas décadas, a população urbana vem crescendo em grande parte dos países. Em 1950, 64% da população americana vivia em zona urbana; em 2010, 80%. Uma projeção de 2014, do Departamento de Assuntos Econômicos e Sociais das Nações Unidas, estima que até 2050 este crescimento se manterá. A taxa de população urbana de países como Brasil, Estados Unidos da América, Canadá e Reino Unido se aproximará de 90% no mesmo ano (UNDESA, 2014).

Este comportamento é caracterizado pelo fato das cidades serem importantes no desenvolvimento social e educacional, pois é no centro urbano que há uma maior concentração de instituições de ensino, empresas e serviços de comércio. Ou seja, a importância dos meios urbanos vem crescendo devido à relação entre as atividades presentes neles e as necessidades das pessoas (MEIJER; GIL-GARCIA; BOLÍVAR, 2016).

O funcionamento de uma cidade resume-se na utilização de recursos para produzir um produto ou serviço. Alguns resultados deste processo causam impactos no meio ambiente, como o descarte de resíduos (ALBINO; BERARDI; DANGELICO, 2015), por exemplo. Problemas ambientais como esse, juntamente com problemas resultantes do aumento populacional urbano, caracterizam os problemas urbanos que os líderes políticos das cidades necessitam resolver.

Portanto, há a necessidade de criação de modelos que ofereçam transporte eficiente, gestão integrada de resíduos e participação efetiva da população, pois estes são importantes para o crescimento de uma cidade (MEIJER; GIL-GARCIA; BOLÍVAR, 2016). O termo *Smart City* (SC), então, refere-se à aquilo que emprega tecnologia para encontrar soluções eficientes para estes serviços.

O termo SC foi utilizado pela primeira vez em 1990, mas se limitava ao uso das Tecnologias da Informação e Comunicação (TIC) na construção civil (ALBINO; BERARDI; DANGELICO, 2015). Somente alguns anos depois que este termo surgiu como um modelo para resolver problemas urbanos e aumentar a qualidade de vida das pessoas que vivem nela. Fortemente aplicada nas soluções que promovem uso consciente de recursos ambientais, SC se relaciona a sustentabilidade (NAM; PARDO, 2011).

As aplicações presentes em SC são variadas, pois podem envolver soluções de qualquer problema característico presente em uma cidade. Elas são comumente categorizadas em *Smart Economy, Mobility,*

Environment, People, Living e Governance.¹ Tais aplicações se fundamentam, principalmente, no processamento de dados acerca da cidade e, portanto, necessitam de uma grande quantidade desses para obter informações precisas. Uma das metodologias utilizadas na junção de dados é o Sensoriamento Participativo (SP).

O SP é um paradigma de aquisição de dados através de dispositivos móveis. Nesta concepção dados gerados pelos sensores de *smartphones*, como localização, em conjunto com dados subjetivos fornecidos pelos usuários são utilizados para obter informações sobre a dinâmica da cidade e como as pessoas interagem sobre ela (CELES et al., 2015).

Neste trabalho é apresentada uma proposta de arquitetura para plataforma de Sensoriamento Participativo em uma SC. Esta arquitetura envolve a coleta de dados acerca do ambiente através do uso de dispositivos móveis dos participantes da aplicação. Os dados recebidos são enviados para armazenamento no *middleware*. Estes dados podem ser, em seguida, processados a fim de gerar informações válidas acerca da dinâmica da cidade. Estas são disponibilizadas para serem utilizadas pelos líderes políticos no processo de alocação de recursos públicos.

1.1 JUSTIFICATIVA E MOTIVAÇÃO

SC é o termo relacionado a projetos que buscam otimizar o uso de recursos públicos, alcançar sustentabilidade e melhorar a qualidade de vida dos cidadãos com o uso da tecnologia (ZANELLA et al., 2014). Como a população urbana vem aumentando nas últimas décadas, inclusive no Brasil, conseqüentemente problemas urbanos cresceram. Portanto, o campo de pesquisa de SC vem se destacando na criação de modelos que amenizem situações negativas relacionadas a envelhecimento populacional e mobilidade, por exemplo.

Dentro de SC, SP é um paradigma que explora a ubiquidade de dispositivos móveis para coletar dados acerca da cidade. Portanto, através de uma Rede de Sensoriamento Participativo (RSP), é possível trabalhar de forma similar à uma Rede de Sensores Sem Fio (RSSF), porém com um custo menor, devido à presença de variados sensores somente em um dispositivo e pelo fato da manutenção dos *smartphones* ser responsabilidade do usuário da Rede.

Pode-se destacar ainda que, a abrangência da RSP é potenci-

¹Tradução: Economia inteligente, mobilidade inteligente, meio ambiente inteligente, pessoas inteligentes, vida inteligente e governo inteligente.

almente maior que a da RSSF, pois o número de dispositivos móveis vem crescendo em diversos países, inclusive no Brasil. A pesquisa de Meirelles (2017) comprova relatando que, até o fim de 2017, o Brasil terá um *smartphone* por habitante, e, de acordo com o IBGE (2016), o celular se tornou o principal meio de acesso à Internet em 2016. Como a quantidade de usuários potencial de uma RSP em uma SC é o número de cidadãos dela que possui *smartphone*, é possível obter uma amostragem maior que a de uma RSSF.

Portanto, explora-se a popularização e os recursos disponíveis de dispositivos móveis, como *smartphones* e *tablets*, na criação de uma arquitetura para RSP no projeto de uma SC. O desenvolvimento de uma arquitetura para uma plataforma transparente e *open-source*, com serviços dirigidos pelos cidadãos, é importante para incentivar pesquisas neste campo a fim de encorajar o uso da tecnologia para melhorar a qualidade de vida das pessoas.

1.2 OBJETIVOS

Nesta seção são apresentados os objetivos geral e específicos do presente trabalho.

1.2.1 Geral

Desenvolver uma arquitetura capaz de identificar os principais problemas de uma cidade com base em dados coletados através de SP numa *Smart City*.

1.2.2 Específicos

1. Implementar um protótipo de aplicativo para Android que permita que o usuário insira dados perceptivos acerca da cidade;
2. Empregar aprendizado de máquina no processamento de dados para entender padrões da cidade;
3. Identificar os problemas mais críticos de uma região a partir dos dados recebidos;
4. Entender a dinâmica de uma cidade a partir dos dados coletados pela aplicação.

1.3 METODOLOGIA

O projeto de desenvolvimento deste trabalho está dividido em cinco etapas. A primeira etapa consiste na pesquisa bibliográfica acerca de SC e SP. Nesta etapa, o foco é estudar o contexto dessas pesquisas, bem como as iniciativas já desenvolvidas e em desenvolvimento. Ainda neste estágio, técnicas de desenvolvimento são estudadas, com foco nas principais tecnologias e ferramentas empregadas em aplicações de SP para SC.

A segunda etapa consiste no estudo do modelo e desenvolvimento do protótipo do aplicativo; e na implementação da arquitetura de SP, com apoio da plataforma InterSCity.

Na terceira etapa, os dados para testes são gerados. Tais dados consistem em ocorrências registradas acerca da cidade, bem como dados de localização coletados pelos dispositivos móveis. A elaboração dos dados ocorrerá paralelamente com a parte final da segunda etapa, a fim de garantir compatibilidade.

Na quarta etapa, são realizados os testes com os dados gerados na etapa anterior. Ainda neste estágio, eventuais reparações no sistema implementado serão realizadas.

Na quinta etapa, é realizada a análise do projeto a partir dos resultados obtidos na etapa anterior.

1.4 ORGANIZAÇÃO DO TRABALHO

Este trabalho contém 7 capítulos, contando com esta Introdução.

O **Capítulo 2** descreve as discussões atuais na academia acerca de SC. Também são apresentadas as definições, as características de uma SC e as classificações das aplicações.

O **Capítulo 3** discorre sobre o paradigma de SP, as principais motivações para explorá-lo, bem como os maiores desafios no desenvolvimento de um sistema deste tipo. Algumas iniciativas e estudos feitos nessa área são citados no fim do capítulo.

O **Capítulo 4** apresenta a arquitetura proposta, bem como uma definição das partes envolvidas, com a descrição da comunicação entre o aplicativo Android e o *middleware* e do processamento dos dados.

O **Capítulo 5** descreve detalhadamente a aplicação desenvolvida e seu funcionamento com o apoio da plataforma InterSCity.

O **Capítulo 6** apresenta os resultados de funcionamento da aplicação, as informações obtidas após o processamento dos dados e sua

viabilidade de uso.

O **Capítulo 7** apresenta as considerações finais e também sugestões para trabalhos futuros.

O apêndice apresenta os códigos utilizados para o envio de dados e para os testes realizados na fase de processamento de dados.

O anexo apresenta a documentação da plataforma InterSCity.

2 FUNDAMENTOS DE *SMART CITY*

O termo *Smart City* refere-se ao uso das TICs nos diversos âmbitos de uma cidade, com a finalidade de tornar a alocação de recursos mais eficiente, melhorando a qualidade de vida dos cidadãos. Neste capítulo é feita uma apresentação no que consiste à SC e suas características, das discussões presentes na academia em torno de sua definição, das aplicações e tecnologias utilizadas.

2.1 DEFINIÇÃO DE *SMART CITIES*

A maioria de população vive em cidades na zona urbana. Isso ocorre principalmente porque as cidades são locais importantes para o desenvolvimento econômico e educacional de uma região. Segundo Caragliu, Del Bo e Nijkamp (2011), 75% da população americana vivia na zona urbana em 2000, contrastando com 5,1% em 1790.

A partir destes dados, observa-se que há um crescimento no êxodo rural, migração da região rural para centros urbanos. Este fenômeno não vem acontecendo isoladamente nos Estados Unidos e tem relação com o crescimento populacional global.

Dados das Nações Unidas citam que em 1950 a população mundial era de 2,5 bilhões de pessoas (United Nations, 2013). Também no mesmo documento é previsto que a população alcance 8,1 bilhões de pessoas em 2025 e, em 2050, 9,6 bilhões. A população urbana também cresce a medida que a população aumenta. Portanto, com esta projeção, pode-se dizer que a quantidade de indivíduos morando nas cidades nos próximos anos tenderá a aumentar.

A importância crescente dos meios urbanos acontece porque o desenvolvimento sócio-econômico está fortemente relacionado às atividades presentes nestes meios e, conseqüentemente, às necessidades das pessoas. Portanto, modelos que buscam transporte eficiente, gestão integrada de resíduos e participação efetiva da população, por exemplo, são importantes para o crescimento organizado e sustentável da cidade (MEIJER; GIL-GARCIA; BOLÍVAR, 2016).

Aliando esta necessidade com o uso da tecnologia, estudos sobre SC vem sendo feitos nos últimos anos. Segundo Ballas (2013), esta noção é adotada para buscar desenvolvimento sustentável, crescimento econômico e melhora na qualidade de vida dos cidadãos.

Muitos desses modelos, fazendo uso das Tecnologias da Informa-

ção e Comunicação (TIC), são comumente divididos em subáreas da SC. Entretanto, ela não se limita somente ao uso das TICs, visto que a gama de aplicações presentes nesse ramo de estudo é ampla.

Na literatura, ainda não há um consenso sobre no quê o termo SC consiste. Albino, Berardi e Dangelico (2015) sustentam a incerteza em relação ao termo assertando que SC é um termo impreciso que é usado em contextos diferentes, que nem sempre estão relacionados às cidades que possuem características inteligentes.

Além disso, definições de termos similares se sobrepõem ao de SC, tornando difícil obter uma visão consistente deste. Nam e Pardo (2011) reforçam este fato afirmando que, por estar sendo usado popularmente em diferentes contextos, há uma variedade conceitual com a substituição de *smart* por outros adjetivos, como *intelligent* ou *digital*.

Uma cidade digital busca um ambiente que possibilite o compartilhamento de informação e a colaboração mútua (ALBINO; BERARDI; DANGELICO, 2015). A conexão entre integrantes da rede de dados é fundamental. Diferentemente da SC, ela não realiza inferências com as informações que circulam e não há planejamento a partir delas.

Segundo Komninos, Pallot e Schaffers (2013), cidades inteligentes são aquelas que unem a inteligência humana com o compartilhamento de informação num ambiente de inteligência coletiva dentro da cidade. Uma cidade inteligente, assim como a cidade digital, faz uso das TICs, porém, neste caso, as informações coletadas por meio dela são estudadas para efetuar mudanças.

Ubiquitous city, outro termo semelhante, tem como característica uma cidade que disponibiliza um ambiente para seus cidadãos que permita o acesso a diferentes serviços através de qualquer dispositivo, independente do local que ele esteja (ALBINO; BERARDI; DANGELICO, 2015).

Nam e Pardo (2011) ainda listam outros conceitos similares à SC, variando de acordo com o que eles chamam de “dimensões”, conforme dados da Tabela 1. A interação das pessoas nos diversos âmbitos de uma cidade através da tecnologia é a característica que difere SC dos termos relacionados nesta Tabela.

O termo SC foi utilizado primeiramente na década de 1990, mas limitado ao uso das TICs na construção civil (ALBINO; BERARDI; DANGELICO, 2015). Somente alguns anos depois que este conceito surgiu como um modelo para resolver problemas urbanos e aumentar a qualidade de vida das pessoas que vivem na cidade ou usufruem dos recursos dela, relacionando SC com sustentabilidade (NAM; PARDO, 2011).

Zanella et al. (2014) adotam uma definição semelhante com à de

Tabela 1 – Conceitos relacionados à SC

Dimensão	Conceito
Tecnologia	Digital city
	Intelligent city
	Ubiquitous city
	Wired city
	Hybrid city
	Information city
Pessoas	Creative city
	Learning city
	Humane city
	Knowledge city
Comunidade	Smart community

Adaptado de: Nam e Pardo (2011)

Nam e Pardo (2011), ele sustenta que SC é aquilo relacionado a oferecer um serviço público com mais qualidade aos cidadãos, bem como utilizar os recursos mais eficientemente, de modo que atenda às necessidades das pessoas, e que o custo de operação seja reduzido.

Um fator que acentua a discrepância entre as definições existentes é a manobra de *marketing*, que utiliza o termo como um rótulo, ao invés de realmente empregar tecnologia a fim de obter conclusões realmente inteligentes (HOLLANDS, 2008).

Entretanto, é possível encontrar definições próximas na literatura. Segundo Xiao, Lim e Ponnambalam (2017), SC adotam TICs modernas, como a *Internet of Things*² (IoT) e computação em nuvem para resolver questões econômicas, sociais, de energia, ambientais e governamentais.

Outros autores, por outro lado, vão além na definição da palavra *smart* e afirmam que uma cidade é inteligente quando há investimento em capital humano e social, infraestrutura empregando modernas Tecnologias da Informação e Comunicação, base para o crescimento econômico e de qualidade de vida, com gestão inteligente de recursos naturais, fazendo uso da democracia eletrônica (e-democracia) (CARAGLIU; Del Bo; NIJKAMP, 2011).

Correlacionando com a questão do crescimento populacional le-

²Tradução: Internet das Coisas

vantado anteriormente, Chourabi et al. (2011) afirma que SC vem se tornando importante porque fazer uma cidade ser inteligente é uma estratégia para responder problemas urbanos devido ao crescimento da população neste meio. Esses podem ser alta densidade populacional, envelhecimento populacional e consequências da rápida urbanização.

2.2 CARACTERÍSTICAS DE UMA *SMART CITY*

Por ainda não possuir um conceito bem definido e pela vasta abrangência de aplicações, é possível encontrar na literatura algumas características que ajudam na identificação de SC.

A automatização de funções rotineiras a fim de melhorar e facilitar o processo de atividades para as pessoas, bem como tornar a cidade eficiente por meio de monitoramento e análise de dados fazem parte de SC (BATTY et al., 2012).

Apesar das TICs serem fundamentais na execução das atividades da SC, elas são consideradas como um meio por alguns autores, como Caragliu, Del Bo e Nijkamp (2011) e Albino, Berardi e Dangelico (2015). Em contrapartida, outros autores focam principalmente no uso das TICs. O aspecto tecnológico é fundamental, juntamente com o espaço cibernético e o físico. Sendo que o primeiro consiste na infraestrutura, no armazenamento dos dados e no funcionamento das aplicações (KOMNINOS; PALLOT; SCHAFFERS, 2013).

Para Caragliu, Del Bo e Nijkamp (2011) a SC é representada por seis características principais:

1. A presença de uma infraestrutura que permita a existência de uma rede capaz de melhorar a economia, promover o desenvolvimento social e propiciar a participação política. Dispositivos móveis, telefones, serviço de internet e satélites tornam isso possível.
2. O desenvolvimento urbano dirigido ao crescimento da economia que, por influenciar no índice de satisfação econômico, incentiva o surgimento de novos negócios e inovações.
3. A participação das pessoas de todas as classes é encorajada, promovendo a inclusão social através da tecnologia.
4. Há estímulo da participação de empresas de tecnologia no crescimento urbano, pois aumenta o capital de giro, que tem relação com o crescimento populacional e, conseqüentemente, econômico.

5. Investimento no crescimento urbano a partir dos dados sociais e econômicos. A SC reconhece que a população deve ser capacitada tecnologicamente e incentivar o aprendizado.
6. Promove o descarte inteligente de recursos e viabiliza a sustentabilidade ambiental, pois percebe que o crescimento populacional e a falta de planejamento pode levar à escassez de recursos.

Similarmente, Albino, Berardi e Dangelico (2015) listam as seguintes características:

- Uma infraestrutura tecnológica interligada que permite eficiência política, desenvolvimento social e cultural;
- Crescimento urbano a partir de incentivo à empresas;
- Inclusão social;
- Planejamento do meio ambiente.

Batty et al. (2012) também não dão destaque às TICs, mas sim ao modo como são utilizadas. Se houver funções inteligentes que integrem os dados de uma cidade para uma finalidade, com o propósito geral de melhorar a qualidade de vida, o funcionamento de serviços e a dinâmica da cidade, então ela é considerada inteligente. A Figura 1 ilustra o relacionamento entre os objetivos característicos de uma SC e suas aplicações, discutidas detalhadamente na Seção 2.3.

Nam e Pardo (2011) categorizam as características segundo os aspectos tecnológico, humano e institucional. O aspecto tecnológico é crucial em SC para processamento e armazenamento de dados, percepção em tempo real e para fazer decisões mais inteligentes porque consideram diferentes informações simultaneamente com maior precisão. O humano caracteriza-se pelo crescimento mútuo através da colaboração comunitária, usando pensamento criativo e explorando a tecnologia na educação. Dando suporte a esses dois aspectos, o suporte do governo e o engajamento dos líderes políticos é importante para o sucesso. Fatores intitucionais conectam os envolvidos, a saber, cidadãos, empresas e outras organizações. A transparência das decisões, viabilizando a participação das pessoas através de plataformas tecnológicas também são cruciais nesta questão.

Além disso, conforme discutido anteriormente, as pessoas são o fator diferencial em SC e, portanto, a participação dos cidadãos é ampliada com o uso da tecnologia através das redes sociais ou de aplicações específicas, caracterizando o SP.

Figura 1 – Componentes característicos de uma SC.



Fonte: Adaptado de Gharaibeh et al. (2017).

2.3 APLICAÇÕES

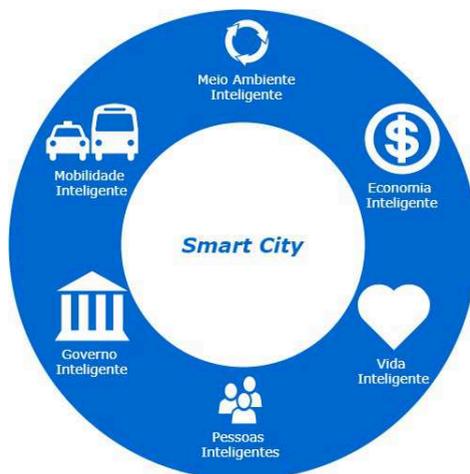
Visto que a área de SC é bastante abrangente, suas aplicações são comumente subdivididas em subáreas. Apesar de haver algumas diferenças entre as subdivisões feitas por alguns autores, há uma conformidade na divisão de suas principais subáreas.

Zanella et al. (2014) adere aos setores adotados pelo projeto *European Smart Cities*: governo inteligente, mobilidade inteligente, serviço público inteligente, construção inteligente e meio ambiente inteligente. A Figura 2 ilustra os setores das aplicações.

Outros autores simplificam ainda mais essa subdivisão. Bățăgan (2011), foca sua pesquisa em quatro áreas de estudo, chamadas de sistemas centrais da SC: sistema de transporte, sistema de saúde, sistema de governo e sistema de educação.

Entretanto, a maior parte das pesquisas na área de SC divide as aplicações em seis principais subáreas. Albino, Berardi e Dangelico (2015), Nam e Pardo (2011) e Caragliu, Del Bo e Nijkamp (2011)

Figura 2 – Tipos de aplicações de SC.



Fonte: Elaborado pelo autor.

as citam: economia inteligente, mobilidade inteligente, meio ambiente inteligente, pessoas inteligentes, vida inteligente e governo inteligente. Adotando essa conformidade da maior parte dos autores, elas são expandidas detalhadamente nas próximas seções.

2.3.1 Economia Inteligente

Albino, Berardi e Dangelico (2015) esclarecem que o conceito de Economia Inteligente está relacionado com uso das TICs na indústria. Giffinger (2007) afirma que este termo está relacionado ao uso inteligente de recursos, à iniciativa de negócios criativos e inovadores que atraem mais investimentos e trazem desenvolvimento, aumentam a produtividade e estimulam a competitividade econômica.

Esta área ainda pode explorar locais de marketing e dados de rastreamento, a fim de obter informações sobre o comportamento do consumidor. Estudos neste ramo procuram identificar a probabilidade de um indivíduo comprar determinado produto, dado um trajeto percorrido (PAN et al., 2013).

Visto que a SC foca no compartilhamento de informações e na

melhora da qualidade de vida da comunidade, a cidade pode incentivar serviços como ambientes de empreendimento compartilhados, que visam o começo de novos negócios (ECKHOFF; WAGNER, 2017).

2.3.2 Mobilidade Inteligente

Eckhoff e Wagner (2017) ressaltam que Mobilidade Inteligente é uma das áreas mais associadas à SC, pois é um problema em muitas cidades e aplicativos vem sendo utilizados para amenizar o congestionamento no trânsito. Essa área de SC objetiva melhorar a qualidade de vida das pessoas através do estudo do fluxo de veículos a fim de aumentar a segurança e diminuir o tempo gasto no trânsito.

Portanto, soluções que disponibilizem acessibilidade e que façam uso das TICs e promovam transporte seguro são relacionadas à Mobilidade Inteligente (GIFFINGER, 2007). Bătăgan (2011) destaca que soluções a partir do estudo da mobilidade de uma cidade pode diminuir a emissão de gases, unificar pagamentos relacionados a transporte e aumentar o uso de transporte público.

Segundo Su, Li e Fu (2011), um sistema inteligente de controle de tráfego pode aprimorar o trânsito através de leituras de fluxo de automóveis e pessoas e previsões de congestionamento. Pessoas e veículos podem atuar como sensores na rede de mobilidade disponibilizando em tempo real dados relacionados ao trânsito - congestionamento, desvio de trajeto e colisões, por exemplo (PAN et al., 2013).

Altomare et al. (2013) resume que com a Mobilidade Inteligente será possível conhecer inúmeros trajetos em consequência da análise de padrões e comportamentos feito com dados extraídos do rastreamento de sensores da rede.

2.3.3 Meio Ambiente Inteligente

Conforme Giffinger (2007), Meio Ambiente Inteligente se refere a um melhor aproveitamento dos recursos naturais, a planejamentos que pretendem a diminuição da emissão de gases poluentes, e à proteção do meio ambiente com uma gestão sustentável de recursos naturais.

Juntamente com a gestão dos recursos ambientais, investimentos em parques e tratamento de água e esgoto são projetos que tem impacto positivo na conscientização social de sustentabilidade (CHOURABI et al., 2011).

Através da criação de mapas da qualidade do ar, por exemplo, é possível verificar áreas de condições críticas para a saúde. Isso permite as autoridades a decidirem a necessidade de um plano de evacuação ou de aviso à população (ECKHOFF; WAGNER, 2017).

Segundo Su, Li e Fu (2011), este tipo de aplicação faz uso abundante de sensores, tecnologia, dispositivos de identificação e automação a fim de proporcionar um ambiente mais confortável que também economize energia. Sistemas como esse possuem funcionalidades como iluminação automática, controle de temperatura personalizado e *smart grid*.³

2.3.4 Pessoas Inteligentes

Está relacionada à aplicações de valores sociais e humanos. Giffinger (2007) aponta mentalidade aberta, flexibilidade, envolvimento na vida comunitária, criatividade e aprendizado como itens desta área. Estudos nestas aplicações medem envolvimento em trabalho voluntário, nível de recepção de imigrantes e valorização da vida acadêmica, por exemplo.

A vida em comunidade idealizada, fortemente sustentada pela interação através das redes sociais, cria novas redes de contatos por intermédio de aplicações deste tipo. Como podem haver plataformas específicas de acordo com a finalidade, a *Human-Computer Interaction*⁴ (HCI) e a questão do compartilhamento de informações pessoais em tais ambientes comunitários virtuais são delicadas, pois envolvem itens de privacidade (BATTY et al., 2012).

2.3.5 Vida Inteligente

Eventos culturais, saúde de qualidade, segurança, acesso à educação e turismo são áreas relacionadas à Vida Inteligente. Portanto, as métricas que definem se uma aplicação se enquadra em Vida Inteligente são: quantidade de visitas a museus por habitante, expectativa de vida, taxa de criminalidade, taxa de conclusão do ensino superior, quantidade de cômodos na residência e quantidade de hotéis (GIFFINGER, 2007).

O aprimoramento da qualidade do sistema de saúde é uma das pesquisas mais fortes neste ramo. Históricos de saúde eletrônicos e

³Sistema destinado a usar energia elétrica mais eficientemente pelo uso das TICs.

⁴Tradução: Interação Humano-Computador

sistema unificado de médicos, planos de saúde e pacientes são algumas das principais iniciativas (BĂTĂGAN, 2011).

Segundo Pan et al. (2013), uma SC deve possuir um sistema inteligente de segurança. Ele precisa ser capaz de detectar irregularidades na rotina das pessoas, perceber agrupamentos suspeitos e rastrear criminosos. Identificação de pessoas, possíveis criminosos ou pessoas perdidas, por exemplo, são necessários na garantia de qualidade de vida.

Reforçando esta concepção, Altomare et al. (2013) diz que vigilância, detecção de incêndio e controle de acesso devem ser instalados em prédios públicos e comerciais.

2.3.6 Governo Inteligente

Conforme defendido por Baptista (2005), Governo Inteligente envolve diversos aspectos da vida pública. São objetivos deste setor: melhorar a gestão pública, empregar tecnologia nos serviços prestados, descentralizar as decisões de caráter público e obter progresso através do uso de novas tecnologias.

Bătăgan (2011) ainda relaciona processamento inteligente de documentos, considerando prioridade e início do processo e alocação de recursos públicos como aplicações nesta área.

A partir da concepção destes projetos, surge a tendência denominada *e-Governance* ou *e-Democracy*⁵, conjunto de serviços prestados em plataforma *online*. Albino, Berardi e Dangelico (2015) definem este conceito dentro de Governo Inteligente: *e-governance* é essencial quando se trata da aproximação do cidadão e das iniciativas de uma cidade inteligente. Além disso, a participação das pessoas e o acesso à informação torna o processo transparente, fator muito importante quando se trata de recursos públicos.

Eckhoff e Wagner (2017) afirmam que o propósito de habilitar tais plataformas é permitir que o cidadão se envolva nas decisões públicas e que haja envolvimento dele no desenvolvimento da cidade.

2.4 TECNOLOGIA

As TICs são consideradas um dos principais fatores que tornam as SCs possíveis. Elas são capazes de mudar o comportamento das

⁵Tradução: Governo eletrônico ou democracia eletrônica

pessoas, transformando a vida profissional e a pessoal, de aprimorar o conhecimento e de ajudar a gerar inovações (HOLLANDS, 2008).

As tecnologias empregadas nas SC podem ser vistas através de dois aspectos: técnico e social. O primeiro diz respeito à infraestrutura da cidade: cabos, sensores, métricas de desempenho e redes; o segundo, a tipos de informação e redes de conhecimento geradas pela população (HOLLANDS, 2008).

Nam e Pardo (2011) dão ênfase ao uso de tecnologias como uma integração entre sistemas, infraestrutura e serviços. As TICs são ferramentas que devem ser utilizadas para criar um ambiente que seja inovador, permitindo a interação entre as pessoas, e que o conhecimento proveniente desta permita que a criatividade seja explorada ainda mais em outras inovações.

Coleta e processamento de dados são processos fundamentais em SC que necessitam de tecnologia para serem realizados. Com o intuito de aprimorar o tráfego, diminuir congestionamentos através de rotas alternativas é um dos destaques de aplicação. Necessária para aprender rotas mais utilizadas, caminhos alternativos e, portanto, estudar comportamentos de mobilidade, a aquisição de dados de deslocamento por dispositivos móveis também é importante (PAN et al., 2013).

Dentre as tecnologias utilizadas em SC, destaca-se a computação na nuvem. Ela se relaciona com os dispositivos que fazem a aquisição de dados, tais como sensores, *smartphones* e dispositivos móveis com permissão de coletá-los do ambiente para produzir conhecimento. Altomare et al. (2013) defende o uso da computação na nuvem mencionando projetos que empregam esta tecnologia, como o *Living Lab*, da *European Platform for Intelligent Cities* (EPIC) e o *Smarter City Solutions*, da IBM.

A aquisição dos dados ocorre por meio de uma rede de sensores interconectados na Internet (MITTON et al., 2012). Sensores capazes de detectar a localização estão presentes em diversos dispositivos atualmente. Pan et al. (2013) divide tais aparelhos em quatro categorias: dispositivos móveis, veículos, cartões inteligentes e sensores flutuantes.

O uso de dispositivos móveis em tais áreas apresenta grande potencial, pois a maioria das pessoas possui no mínimo um *smartphone* (VAIDYA et al., 2015). Ainda nesta categoria pode-se ressaltar o uso de *tablets*. Tais aparelhos estão equipados com GPS⁶ (Global Positioning System), *bluetooth* e *Wi-Fi*, que permitem determinar a localização do dispositivo. Além dessas tecnologias, também é possível obter a posição por meio de cálculos na rede de telefonia (PAN et al., 2013).

⁶Sistema de Posicionamento Global

Veículos de última geração também vem com GPS e, neste caso, podem fornecer grande quantidade de dados de rastreamento, tanto pela rota adotada quanto pelos destinos (PAN et al., 2013).

Cartões inteligentes destinam-se à união de mais de um documento em um só e funcionam de acordo com o padrão ISO/IEC 14443. Podem efetuar pagamentos e destinar-se à identificação. Os locais onde foram utilizados servem como estação de localização (ECKHOFF; WAGNER, 2017).

Por último, sensores flutuantes são objetos com etiquetas de Identificação por Radiofrequência (RFID) incorporadas (PAN et al., 2013). Através de sinais de rádio da base transmissora, estas etiquetas respondem informando a posição atual dos dispositivos que usam esta tecnologia.

2.4.1 *Middleware*

Em SC, há a agregação de diferentes aplicações, com um alto volume de dados, funcionando e se comunicando com um centro de controle. Esta comunicação estabelecida entre os dispositivos e o centro deve ser coordenada e eficiente. O *middleware* é considerado importante porque é uma solução para esta e outras situações, como escalabilidade, heterogeneidade de dispositivos e análise de informações (SMARTMETROPOLIS, 2016).

Segundo Couloris, Dollimore e Kindberg (2007), o *middleware* é um modelo computacional que oferece processos e objetos que implementam comunicação e dão suporte à aplicações distribuídas e a compartilhamento de dados. E, portanto, ele funciona como uma camada de abstração entre a aplicação e o dispositivo.

Devido à bibliotecas e ferramentas disponibilizadas pelos *middlewares*, a produtividade no desenvolvimento cresce, sendo fundamental para o sucesso de SC (VILLANUEVA et al., 2013).

Segundo SmartMetropolis (2016), os requisitos fundamentais de *middleware* são:

- Interoperabilidade;
- Descoberta e gerenciamento de dispositivos;
- Adaptação dinâmica;
- Ciência de contexto;
- Escalabilidade;

- Tratamento de grandes volumes de dados;
- Segurança.

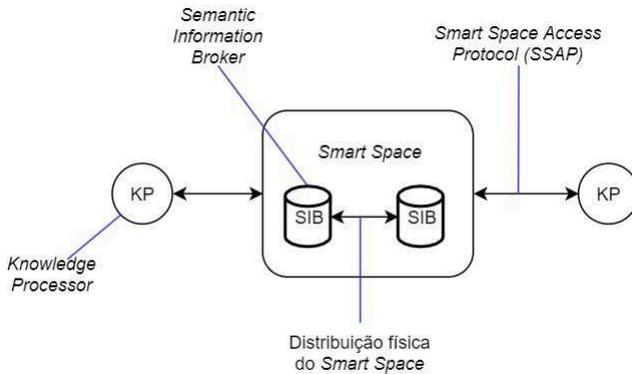
Há diversas plataformas de *middleware* disponíveis para IoT. Algumas delas são apresentadas nas seções a seguir.

2.4.2 SOFIA

SOFIA, acrônimo de *Smart Objects for Intelligent Applications*, é uma arquitetura de *middleware* que permite a interoperabilidade de vários dispositivos e sistemas. É *open-source*, multiplataforma, multi-linguagem e implementa os protocolos de comunicação TCP, MQTT, HTTP⁷ e outros (VILLANUEVA et al., 2013).

Foi desenvolvido originalmente pelo projeto Artemis e, após continuação da Indra, a plataforma empresarial SOFIA2 foi criada. A Figura 3 ilustra a arquitetura da plataforma SOFIA2.

Figura 3 – Arquitetura da Plataforma SOFIA2.



Fonte: Adaptado de: Indra (2014).

O *Smart Space* é o ambiente virtual onde as aplicações interoperam. Seu núcleo é composto por um, ou mais, *Semantic Information Broker* (SIB). É nele que as informações são recebidas, processadas e

⁷Do inglês: *Transmission Control Protocol, Message Queuing Telemetry Transport e Hypertext Transfer Protocol*, respectivamente.

armazenadas. Através do SIB, o *Knowledge Processor* (KP), que representa a aplicação, interopera com o *Smart Space*. Entre eles, há o *Smart Space Access Protocol* (SSAP), que é o protocolo de mensagem para comunicação (INDRA, 2014).

O SOFIA utiliza ontologias, que são as entidades de informação do sistema, representadas no formato JSON-Schema.

2.4.3 FIWARE

FIWARE é uma plataforma *open-source* e escalável para gerenciar aplicações de conteúdo heterogênea. Tem como objetivo disponibilizar uma plataforma genérica para serviços de IoT através de interfaces de programação (API), tradução de *Application Programming Interface*. É composta por uma biblioteca de componentes chamados *Generic Enablers* (GE), fundamentais para a composição do FIWARE (SMARTMETROPOLIS, 2016).

Tabela 2 – Principais recursos dos GEs.

GE	Recursos
<i>Applications, Services and Data Delivery</i>	Gerenciamento de aplicações, serviços e dados
<i>Data/Context Management</i>	Manipulação de grande quantidade de dados, gerando conhecimento
<i>Interface to Networks and Devices (I2ND)</i>	Integração de <i>middleware</i> , redes e dispositivos robóticos com os GEs
<i>Internet of Things (IoT) Services Enablement</i>	Conectividade de dispositivos
<i>Cloud Hosting</i>	Infraestrutura de hospedagem de computação na nuvem
<i>Security</i>	Disponibiliza mecanismos que cumprem requisitos de segurança
<i>Advanced Web-based User Interface</i>	Melhora a experiência do usuário na aplicação

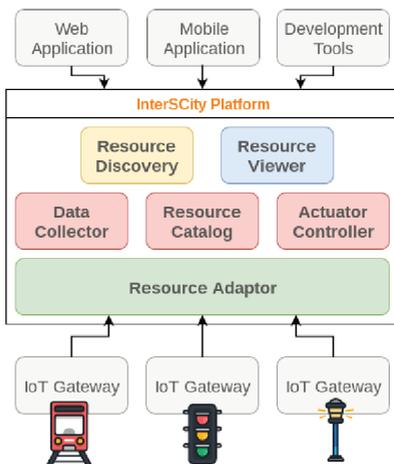
Fonte: Elaborado pelo autor.

Os GEs disponíveis são categorizados em *Data/Context Management*, *Internet of Things Services Enablement*, *Advanced Web-based User Interface*, *Security*, *Interface to Networks and Devices (I2ND)*, *Architecture of Applications/Services Ecosystem and Delivery Framework* e *Cloud Hosting*. Essas categorias também são apresentadas como componentes dos capítulos técnicos da arquitetura do FIWARE. Seus principais recursos e componentes são apresentados na Tabela 2.

2.4.4 InterSCity

Sendo desenvolvida pelo *Smart City Research Group* da Universidade de São Paulo (USP), a plataforma InterSCity é *open source* e baseada em microsserviços que oferecem apoio ao desenvolvimento de aplicações para SC (M. Del Esposte et al., 2017).

Figura 4 Arquitetura da plataforma InterSCity.



Fonte: M. Del Esposte et al. (2017).

No seu estado atual, suporta os principais tipos de aplicações de SC - transporte público, segurança pública e monitoramento ambiental - através de serviços que gerenciam recursos e armazenamento de dados. Sua arquitetura é composta por seis microsserviços (Figura 4) que, em conjunto, provém o serviço de suporte para aplicações de SC. De acordo com M. Del Esposte et al. (2017), o microsserviço *Resource Adaptor*

integra dispositivos; os *Resource Catalog*, *Data Collector* e *Actuator Controller* gerenciam recursos e dados; o *Resource Discovery* procura recursos por contexto; e o *Resource Viewer* provê a visualização.

3 SENSORIAMENTO PARTICIPATIVO

O Sensoriamento Participativo (SP) caracteriza-se pela captação de dados através de dispositivos pessoais dos participantes, bem como a opinião e observações deles acerca de experiências vividas. Neste capítulo, é discutida a definição de SP, suas características, bem como a descrição de algumas iniciativas nesta área.

3.1 SENSORIAMENTO DISTRIBUÍDO

O surgimento das Redes de Sensores Sem Fio (RSSF) fez pesquisadores visionarem o uso difundido de sensores (CRISTOFARO; SORIENTE, 2013). Porém, por uma perspectiva de gerenciamento, numa RSSF, cada sensor tem um grau de importância e, caso ele falhe, deve ser recuperado para garantir que poucos dados deste nó sejam perdidos e, assim, assegurar a confiabilidade e a completude dos dados processados pelo operador.

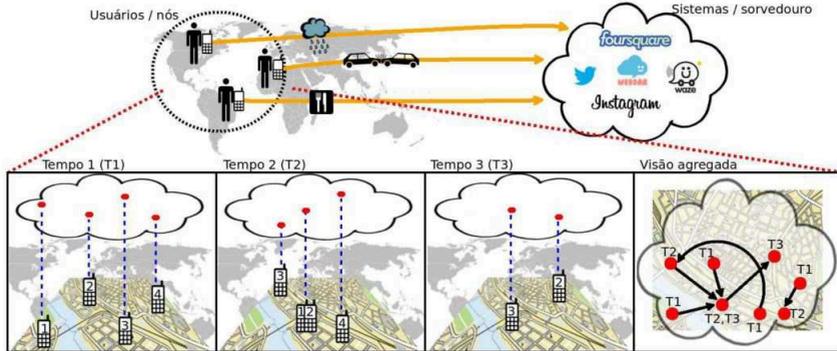
O sensoriamento é um conjunto de técnicas utilizadas para coletar dados. Esses variam de acordo com o sensor empregado e a aplicação. Quando há leitura de vários sensores de diferentes nós numa só rede, sem uma unidade central, caracteriza-se o sensoriamento distribuído. Este pode ser cooperativo, que consiste na colaboração mútua de dados acerca do ambiente (BAZZO; MELGAREJO; YOSHIMURA, 2013).

Os dispositivos móveis são potenciais sensores para uma rede distribuída de sensoriamento, pois são capazes de capturar imagens, checar localização, transmitir áudios e comunicar-se entre si (BURKE et al., 2006). Explorando a oportunidade de usar *smartphones* como sensores em uma área mais abrangente que a de uma RSSF usual, e aproveitando o custo reduzido, pois não há necessidade de instalação física de sensores, surgiu o paradigma de SP. Com a popularização dos *smartphones* e o crescente grau de conectividade entre os usuários, este paradigma tornou-se factível.

Numa RSP, cada pessoa que possui um *smartphone* pode ser um usuário de uma rede colaborativa. Portanto, o sucesso de sistemas de SP deve-se à contribuição dos participantes (ALSWAILIM; ZULKERNINE; HASSANEIN, 2014). Os dispositivos dos usuários participantes funcionam como sensores móveis, diferentemente da RSSF, cujos sensores são fixos. A Figura 5 ilustra o funcionamento de uma RSP, no qual dados de tempos diferentes são coletados e enviados à aplicação que, por sua

vez, irá agregá-los.

Figura 5 – Funcionamento de uma RSP.



Fonte: Extraído de Celes et al. (2015).

Visto que uma das principais características de uma RSP é o uso do dispositivo pessoal do usuário, a manutenção dos componentes físicos que a integram não é responsabilidade da aplicação, mas sim do usuário. Isto torna o custo da RSP mais baixo do que o de uma RSSF. Essa mesma característica, porém, inibe a participação de algumas pessoas, pois a garantia de privacidade é considerada um ponto crítico já que o *smartphone* acompanha o indivíduo e informações pessoais podem ser expostas (CRISTOFARO; SORIENTE, 2013).

Dependendo das informações coletadas, um histórico pode ser manipulado de modo a inferir dados pessoais, independentemente do nível de privacidade escolhido pelo usuário, caso haja essa opção. Algumas dessas informações podem ser barulho, velocidade e localização (SEI; OHSUGA, 2016).

3.2 DEFINIÇÃO DE SENSORIAMENTO PARTICIPATIVO

SP é definido na literatura como um conceito que envolve o uso de dispositivos móveis pessoais, principalmente *smartphones* e *tablets*, como sensores em uma rede cujo funcionamento é similar ao de uma RSSF.

É um paradigma baseado na vasta disponibilidade deste tipo de

dispositivo, que possui sensores como acelerômetro, GPS, giroscópio e magnetômetro, e acesso à Internet (VERGARA-LAURENS; MENDEZ; LABRADOR, 2014).

Celes et al. (2015) simplifica que SP é um processo de coleta de dados distribuídos, sejam eles pessoais ou sensoriados.

Burke et al. (2006) definem SP destacando a forma como os dispositivos móveis são utilizados numa RSP e por quê. Os dispositivos móveis são utilizados a fim de construir uma RSP de forma interativa, que permita a contribuição de qualquer pessoa que participe desta, por meio de coleta e análise de dados locais.

Similarmente, Estrin (2010) define SP como um processo no qual indivíduos e comunidades usam dispositivos móveis e computação na nuvem para coletar e analisar sistematicamente dados para adquirir conhecimento.

Assim como a definição de SC, a de SP ocasionalmente converge para as de outros termos estudados. Segundo Celes et al. (2015), o conceito de SP é similar aos de *Humans as Data Sources* e *Ubiquitous Crowdsourcing*, em que estes são mecanismos de obtenção de conhecimento das pessoas.

Alswailim, Zulkernine e Hassanein (2014) ainda relacionam *mobile sensing*, *opportunistic sensing*, *public sensing* e *crowdsensing* como termos de pesquisa semelhantes ao de SP. Porém, a diferença entre os termos mencionados não é caracterizada pelo conceito em si, mas sim pelo modo de operação.

Os modos de sensoriamento podem qualificar diferentes termos incorporados em SP. Estrin (2010) relaciona três modos de sensoriamento: manual, automático e oportunista. A iniciativa de enviar dados deve partir do próprio usuário no modo manual. Quando automático, a aplicação executa tarefas periódicas que leem e mandam dados, que não necessitam da permissão do usuário. O modo oportunista é orientado a eventos. Sempre que as condições de desencadeamento são satisfeitas, a aplicação é executada. Similar ao modo automático, não necessita da ciência do participante em todo evento.

Para Guo et al. (2014), *Mobile Crowd Sensing* é uma extensão de SP. O primeiro considera todos os dados provenientes dos participantes. Além da contribuição sensorada pelos dispositivos móveis, dados implícitos e explícitos provenientes de redes sociais, por exemplo, também são usados.

3.3 CARACTERÍSTICAS

Quando comparada à RSSF, a RSP se diferencia pelo fato de os nós da rede serem independentes. Ou seja, o custo de manutenção é de responsabilidade do proprietário do dispositivo móvel. Já numa RSSF, os nós são de responsabilidade do gerenciador da rede.

Numa RSP, todavia, apesar de contribuírem enviando dados para a rede, os nós são independentes dela, de modo que eles podem participar de mais de uma rede simultaneamente (ALSWAILIM; ZULKERNINE; HASSANEIN, 2014).

Segundo Khoi, Rodr e Casteleyn (2017), toda aplicação de sensoriamento participativo se caracteriza pela flexibilidade de qualquer indivíduo poder participar e pela possibilidade de encontrar soluções criativas por meio do estudo do ambiente.

De acordo com Celes et al. (2015), as RSPs possuem quatro características principais:

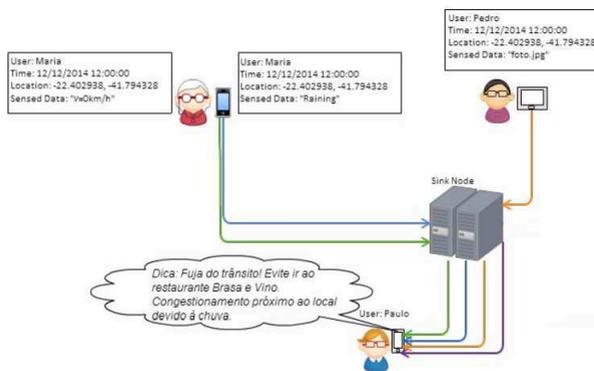
- Cada pessoa com dispositivo móvel é um nó independente;
- Cada nó cobre o seu custo de funcionamento;
- A participação não é obrigatória;
- As limitações de energia dos nós é reduzida.

O funcionamento de uma RSP se inicia com a coleta de dados do ambiente através de dispositivos móveis. Este processo pode ser iniciado pela aplicação, ao chegar em determinado local ou devido à ocorrência de um evento, ou pela detecção do sensor. A seguir, os participantes colaboram enviando dados à central da aplicação que, por sua vez, irá processá-los e analisá-los (ALSWAILIM; ZULKERNINE; HASSANEIN, 2014).

Os dados coletados quando processados devem possuir contexto, pois a credibilidades deles é fundamental na geração de informações confiáveis. Parâmetros como local e tempo são considerados os mais importantes nessa questão, pois quanto mais precisos mais seguras serão as tomadas de decisão (BURKE et al., 2006). Khoi, Rodr e Casteleyn (2017) relaciona a imprecisão e irrelevância de respostas com a falta de contexto dos dados amostrados. A Figura 6 ilustra um exemplo de informação contextual gerada a partir de dados enviados pelos participantes de uma RSP.

As leituras de outros sensores podem ser usadas com a finalidade de validar o contexto. Por exemplo, a direção atual pode verificar

Figura 6 – Geração de informação contextual em uma RSP.



Fonte: Extraído de Freitas (2015).

a integridade de um dado. Porém, este tipo de confirmação não se limita a dados lidos de um só dispositivo, mas também de outros nas proximidades (BURKE et al., 2006).

O contexto é fundamental para a verificação da integridade dos dados, mas podem entrar em conflito com o nível de privacidade escolhido pelo usuário, pois ele pode estar disposto a compartilhar informações desde que não sejam invasivas. Segundo Khoi, Rodr e Casteleyn (2017), a aplicação deve ser flexível em relação a conteúdo sensorado, bem como garantir sua proteção.

Khoi, Rodr e Casteleyn (2017) menciona duas características relevantes em RSP: incentivar os participantes com prêmios pela colaboração e a capacidade de salvar dados caso não haja conexão com a rede no momento do evento para serem enviados quando possível.

Apesar de SP aludir à ideia de que as pessoas físicas contribuem para o funcionamento de uma aplicação, não há restrição quanto à contribuição de pessoas jurídicas. Quando há suporte da tecnologia utilizada, organizações e empresas podem compartilhar informações relevantes. Visto que seus papéis na sociedade são fundamentais, é lógico inferir que dados da indústria são relevantes na tomada de decisão de uma SC (ALSMAILIM; ZULKERNINE; HASSANEIN, 2014).

3.3.1 Desafios

Como as RSPs lidam com um grande volume de dados e de participantes, os desafios das aplicações estão relacionados à privacidade dos usuários, manutenção e processamento dos dados (KANHERE, 2013).

Alswailim, Zulkernine e Hassanein (2014) afirmam que o maior desafio de aplicações de RSP é a privacidade de seus usuários. O conjunto de dados enviados à central, se não protegidos por criptografia, após estudados podem levar à identificação da pessoa que os enviou.

A possibilidade de ter sua identidade ou seu trajeto diário revelados, tornando-se suscetível a situações de violação de segurança, pode desencorajar a participação do usuário em uma aplicação (GROAT et al., 2012).

Informações como localização, hora e áudio são comumente lidos de seus dispositivos. Por isso, há preocupação em relação à privacidade (ALSWAILIM; ZULKERNINE; HASSANEIN, 2014). Ademais, trajetos diários, bem como comportamentais podem ser inferidos a partir de dados de rota. Portanto, a segurança de dados pessoais é fundamental (CHEN et al., 2014).

Amostras incompletas são problemas a serem resolvidos constantemente neste tipo de aplicação. Os usuários podem escolher dar prioridade a outras funcionalidades em seu *smartphone*, limitando o compartilhamento somente quando seus dispositivos estão com bastante carga. Estas situações levam a intervalos sem dados e, conseqüentemente, à informações menos exatas (KANHERE, 2013).

Khoi, Rodr e Casteleyn (2017) ainda relacionam este problema com eventuais perdas de conexão com a Internet. Perda de sinal, bateria esgotada, consumo total dos dados móveis, por exemplo, podem inibir a participação efetiva das pessoas. Portanto, é necessário que o processamento não consuma muita energia e que seja eficiente.

A avaliação de contexto dos dados é um desafio, pois é necessário combinar diferentes atributos para defini-lo. Kanhere (2013) relaciona três etapas para inferi-lo com êxito: coletar dados rotulados apropriadamente, identificar a categoria e escolher o algoritmo de classificação adequado.

Celes et al. (2015) aponta a questão da validade dos dados. Em aplicações que apresentam mudanças significativas ao longo do tempo, os dados são relevantes somente durante um período finito. Caso contrário, podem levar à conclusões incorretas, pois há a possibilidade de não representarem mais a realidade após esse período.

Kanhere (2013) também considera que neste caso é necessário

lidar com a inserção errônea proposital por parte de usuários mal intencionados.

3.3.2 Oportunidades

A grande oportunidade vista em RSP é o uso de dispositivos móveis, que facilitam a obtenção de dados precisos por meio da exploração dos sensores presentes neles, que as pessoas levam consigo diariamente (BURKE et al., 2006).

Burke et al. (2006) menciona as tecnologias e os sensores integrados a dispositivos móveis atualmente: GPS, localização pela antena de telecomunicação e outras tecnologias que podem prover posição através de cálculos e sincronização.

A grande utilização de *smartphones* e *tablets* pelas pessoas juntamente com a presença diária de tais sensores com elas estimulam soluções que explorem esse fato. Segundo Bogdan-Martin et al. (2017), em 2016, 71% de atividades na Internet foram feitas através de dispositivos móveis.

Partindo de tais dados, que encorajam iniciativas que utilizam dispositivos móveis, Celes et al. (2015) afirma que as RSPs podem se tornar uma ferramenta poderosa para entender o relacionamento entre as pessoas através da rede e, a partir disso, compreender o ambiente em que vivem.

Além disso, uma RSSF está sujeita a falhas, uma vez que o seu funcionamento depende da correta coordenação das ações dos seus nós sensores, que possuem severas restrições de energia, processamento e memória. Já as RSPs são formadas por entidades autônomas e independentes, os seres humanos, o que torna a tarefa de sensoriamento mais resiliente a falhas individuais (CELES et al., 2015).

As TICs tornaram possível a presença ubíqua de dispositivos móveis e, por causa disso, sistemas de RSP permitem a junção de dados de pessoas com históricos variados e também de uma grande área (ESTRIN, 2010). O reconhecimento de padrão preciso é possível quando o maior número possível de casos é considerado (BURKE et al., 2006).

Celes et al. (2015) corrobora este fato quando compara a escalabilidade das RSSF e das RSP. Empregar uma RSSF em uma metrópole é mais difícil por questões de custo com aquisição, instalação e manutenção de equipamento. O custo de uma RSP, todavia, é distribuído pois explora a popularização e ubiquidade de dispositivos móveis. O investimento de uma RSSF de mesma escala seria centralizado e muito

mais caro.

3.4 INICIATIVAS DE SENSORIAMENTO PARTICIPATIVO

É possível desenvolver aplicações de SP em diversas áreas. Como SP pode ser considerado um paradigma de aplicações de SC, elas podem se caracterizar como de qualquer subárea discutida na Seção 2.3.

Segundo Kanhere (2013), as aplicações de SP possuem duas categorias: centradas nas pessoas ou centradas no ambiente. A primeira foca em atividades diárias cujo objetivo é estudar o comportamento das pessoas e a segunda busca compreender o meio a partir de dados de um grupo de pessoas com as mesmas características.

Na primeira categoria, pode-se citar aplicações que façam acompanhamento da saúde do usuário. DietSense, por exemplo, é um aplicativo que, através de fotos do prato, avalia a qualidade da alimentação, e estima o desperdício de comida. Por meio de uma aplicação *Python*, dados do GPS e áudio são armazenados no formato *eXtensible Markup Language* (XML) (REDDY et al., 2007).

O aplicativo BikeNet funciona de forma similar, porém destinado à prática do ciclismo. Mapas de acompanhamento são criados através dos dados dos sensores, como localização, velocidade e rota. Informações coletivas de desempenho e tráfego são disponibilizados à comunidade ciclista (EISENMAN et al., 2007). Pode-se citar, ainda, o correlato mais popular: Strava⁸, destinado também a outras práticas esportivas, funciona ainda como uma rede social.

Ainda nesta categoria, o estudo realizado por Mun et al. (2009) resultou em uma plataforma que calcula o impacto ambiental no usuário. O PEIR (*Personal Environmental Impact Report*)⁹ estima a qualidade do ar calculando aproximadamente os gases poluentes do local considerando a localização do *smartphone*, o meio de transporte utilizado e as indústrias das redondezas. Através de um serviço *web* em *Python*, recebe dados do GPS no formato JSON (*JavaScript Object Notation*), que são armazenados em um banco de dados baseado em PostGIS.

O aplicativo ALSTOM destina-se a apontar acidentes de trabalho. O responsável, ao invés de utilizar um formulário de acidente para preenchimento das informações, utiliza o aplicativo. O uso de tal aplicação permite que o órgão responsável identifique com mais facili-

⁸ www.strava.com

⁹ Tradução: Relatório Pessoal de Impacto Ambiental

dade as naturezas de acidente mais frequentes e possíveis medidas que necessitem ser tomadas (HUSSIN et al., 2014).

Dentre as aplicações centradas no ambiente, as que estudam o tráfego do trânsito se destacam. Pode-se citar o aplicativo Waze, no qual as pessoas enviam informações em tempo real do trânsito, procurando melhorar o fluxo de veículos (FAROKHI; SHAMES, 2016). O similar Nericell utiliza os sensores do *smartphone* para verificar a condição da estrada: a presença de buracos, lombadas e tráfego (MOHAN; PADMA-NABHAN; RAMJEE, 2008). Desenvolvido para a plataforma Android, o *CrowdOut* é um serviço executado no celular que visa a segurança do tráfego nas estradas.

Também são categorizadas em centradas no ambiente as aplicações que utilizem o microfone do celular para avaliar a poluição sonora por área. A plataforma *Ear-Phone* parte desse princípio e cria mapas de nível de barulho, a fim de ajudar as pessoas, uma vez que a poluição sonora agravada pode danificar a audição e causar impactos negativos na vida social. Uma aplicação detecta o barulho em determinado local e o armazena num banco de dados (RANA et al., 2010).

Outras pesquisas similares foram desenvolvidas. Com a finalidade de obter uma plataforma de baixo custo e a partir de SP, Maisonneuve et al. (2009) desenvolveram o *NoiseTube* e, do mesmo modo, Ruge, Altakrouri e Schrader (2013) desenvolveram o projeto *SoundOfTheCity*.

Em Natal (RN), o projeto SmartMetropolis da Universidade Federal do Rio Grande do Norte (UFRN) busca desenvolver e aplicar métodos de aplicações de serviços de cidades inteligentes. Duas aplicações desenvolvidas por este projeto são baseadas em SP: *Find Trip Natal* e *Fala Natal*. O primeiro é um aplicativo que auxilia as atividades de turismo na cidade. Funciona como guia e contém informações registradas pelos usuários acerca dos pontos turísticos. O segundo, também um aplicativo, destina-se à comunicação entre a prefeitura, a ouvidoria e a população. Nele, é possível registrar solicitações e denúncias, por exemplo (SMARTMETROPOLIS, 2017).

Entretanto, o SP não se caracteriza somente pela interação de um aplicativo próprio. Um estudo realizado por Long, Jin e Joshi (2012) se baseia nas localizações publicadas no *Foursquare* para descobrir padrões de mobilidade e preferências dos usuários. Um estudo brasileiro também estudou as atividades mais procuradas em dias de semana e finais de semana através de *check-ins* efetuados pelos usuários em redes sociais (SILVA et al., 2013). Informações como esta podem ser utilizadas para melhorar os estabelecimentos de entretenimento ou até determinar

o local mais adequado para um novo negócio, de acordo com o perfil das pessoas que frequentam os estabelecimentos próximos.

Cranshaw, Hong e Sadeh (2012) partiram deste princípio e desenvolveram o projeto *Livehoods*. A partir do processamento de dados das redes sociais, utilizando uma API do Foursquare e de dados públicos do Twitter, características das pessoas são determinadas para construir uma cidade com uma infraestrutura mais inteligente, com áreas destinadas à indústria, comércio e educação.

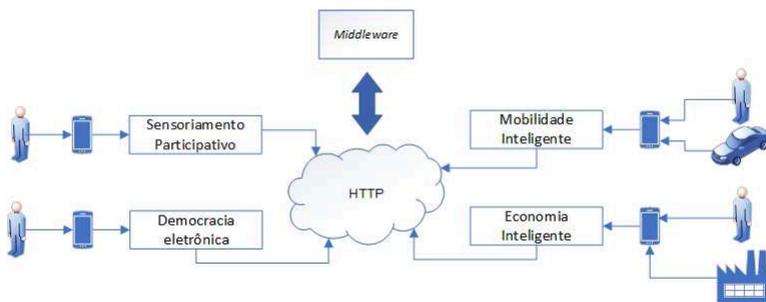
4 ARQUITETURA COMPUTACIONAL PARA SENSORIAMENTO PARTICIPATIVO

Neste capítulo, a arquitetura proposta baseada em SP para SC é apresentada na Seção 4.1. Uma descrição do funcionamento da comunicação entre cliente e servidor é feita na Seção 4.2. O processamento dos dados, bem como sua finalidade, é apresentado na Seção 4.3.

4.1 ARQUITETURA LÓGICA

Esta arquitetura tem o propósito de integrar os dados recebidos por meio de uma, ou mais, aplicações, com o objetivo de identificar características semelhantes entre eles. Ela permite que as pessoas enviem, com seus dispositivos móveis, ocorrências da cidade percebidas por elas, atuando como sensores. Através de aquisição e processamento de dados, a proposta apresentada busca gerar informações relevantes acerca da cidade para auxiliar no gerenciamento desta de acordo com informações fornecidas pelos cidadãos.

Figura 7 – Arquitetura proposta.



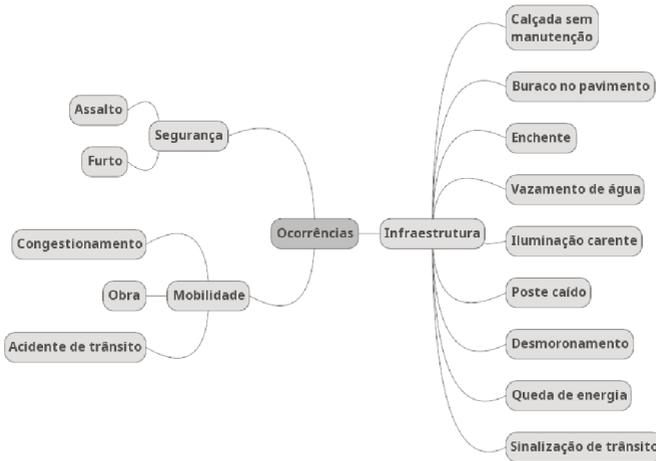
Fonte: Elaborado pelo autor.

Os principais componentes desta arquitetura são os participantes - ou usuários -, o aplicativo e o *middleware*. Os participantes são um conjunto de indivíduos composto por pessoas, indústrias, veículos ou qualquer outro agente envolvido que envie dados acerca da cidade para o sistema. Os aplicativos são quaisquer aplicações voltadas para SC e o *middleware* os oferece serviços. A arquitetura proposta com seus

componentes, replicados em diferentes tipos de aplicações de SC, são ilustrados na Figura 7.

Em democracia eletrônica, os cidadãos opinam sobre possíveis cenários e decisões eminentes em discussão na prefeitura. Em Mobilidade Inteligente, por exemplo, é possível ler dados de localização dos sensores do *smartphone*, a fim de calcular se há congestionamento em determinado trecho, baseado na velocidade média das pessoas nesta área. Em Economia Inteligente, uma possível aplicação dos dados enviados é a avaliação de perfil por região das empresas atuantes. Por meio de um questionário, há a possibilidade de traçar um perfil profissional das pessoas que mais frequentam determinada área.

Figura 8 Tipos de ocorrências.



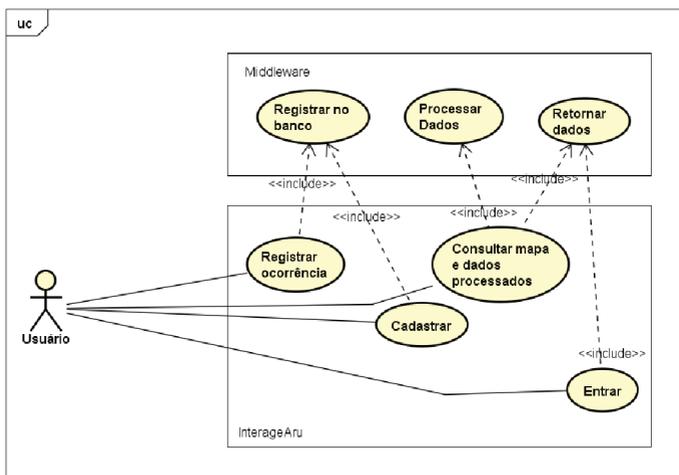
Fonte: Elaborado pelo autor.

Partindo deste princípio, esta arquitetura foi desenvolvida neste trabalho explorando o SP. Os usuários interagem através da aplicação de SP inserindo registros de ocorrências, que serão enviados para o *middleware*. Uma ocorrência é qualquer evento ou acontecimento que esteja relacionado ao funcionamento da cidade. Congestionamento de trânsito, por exemplo, é uma ocorrência relacionada à natureza Mobilidade. Na Figura 8, pode-se observar os tipos de ocorrências que podem

ser reportados no protótipo de aplicativo desenvolvido, bem como suas naturezas: Infraestrututra, Mobilidade e Segurança. Juntamente com a seleção do usuário, a aplicação também envia a data, a hora, e a localização deste registro para o *middleware*.

Conforme descrito anteriormente, a interação dos usuários se caracteriza pelo SP através de dispositivos móveis. Esta arquitetura busca explorar a participação das pessoas a partir de um aplicativo em seus *smartphones*. Aproveitando-se do envio voluntário de dados dos usuários, o aplicativo permite o engajamento cívico. O volume de troca de informações em SP pode ser grande, visto que objetiva englobar os cidadãos de uma cidade. Portanto, a arquitetura proposta deve suportar múltiplos acessos e solicitações.

Figura 9 Diagrama de Caso de Uso



powered by Astah

Fonte: Elaborado pelo autor.

Portanto, a arquitetura proposta deve suportar as funcionalidades ilustradas no Diagrama de Caso de Uso na Figura 9. O usuário, através do protótipo do aplicativo, InterageAru, pode realizar seu cadastro, efetuar o *login*, registrar ocorrências e consultar dados processados. O *middleware*, por sua vez, dá suporte às essas funcionalidades.

4.1.1 *Middleware*

Para fins de realização deste trabalho, as duas primeiras opções discutidas na Seção 2.4.1 não foram utilizadas. O *middleware* FIWARE possui muitas ferramentas de apoio para aplicações em *Smart City*. Entretanto, assim como o SOFIA, é voltado para projetos em equipe. A documentação do primeiro é bastante extensa e, para isso, uma equipe maior seria necessária. O processo de trabalho com a plataforma SOFIA também é mais burocrático, pois demanda formulários de contato e declaração de projeto.

Portanto, a plataforma InterSCity foi a utilizada na realização deste trabalho. Ela é *open-source*, de documentação mais objetiva e atende às principais necessidades da aplicação.

4.2 COMUNICAÇÃO

A fim de estabelecer comunicação entre o usuário e o *middleware*, a aplicação Android funciona como um mediador. A comunicação é baseada no protocolo HTTP, Protocolo de Transferência de Hipertexto¹⁰.

4.2.1 Registro

A plataforma InterSCity, discutida na Seção 2.4.4 é o *middleware* utilizado no desenvolvimento desta arquitetura.

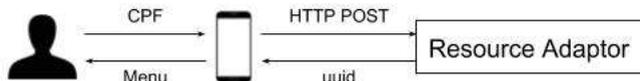
Pelo fato da arquitetura desta plataforma requerer que haja um recurso associado aos dados enviados e com o intuito de poder retornar um histórico de ocorrências registradas pelo usuário, é necessário que ele possua um cadastro.

O registro do usuário na plataforma é realizado de forma simples. No primeiro acesso, a pessoa se registra com o seu CPF. Conforme ilustrado na Figura 10, se o método de solicitação HTTP POST for bem sucedido, o microsserviço *Resource Adaptor* retornará um identificador universal único, *uuid*, em formato JSON¹¹ e, assim, o acesso será permitido. Nos acessos sucessivos, o CPF já cadastrado é informado para ter acesso concedido.

¹⁰Tradução de *Hypertext Transfer Protocol*

¹¹Acrônimo de *JavaScript Object Notation*

Figura 10 – Troca de informações no acesso.



Fonte: Elaborado pelo autor.

Visto que a comunicação com o *middleware* é HTTP, é necessária uma conexão com a Internet para poder utilizar os recursos da aplicação de SP. Portanto, quando este requisito for cumprido, ele enviará uma requisição de acesso com seu *username*, e o servidor responderá a requisição.

Figura 11 – Principais processos entre a aplicação e o *middleware*.



Fonte: Elaborado pelo autor.

Numa aplicação de SP, a comunicação entre as partes do sistema é essencial para o bom funcionamento, pois ela consiste basicamente no envio de informações, sejam elas dados de sensoriamento ou do usuário.

A Figura 11 ilustra as possíveis trocas de informações nesta arquitetura. Em (1) o usuário acessa o aplicativo. Uma nova ocorrência é registrada (2) e enviada ao Banco de Dados (3), que será utilizada na análise dos dados (4). Após solicitação, o usuário recebe um retorno de dados processados (5). Em (6), há a solicitação e retorno de ocorrências registradas, bem como a confirmação de *login* e cadastro.

4.3 PROCESSAMENTO DE DADOS

A etapa de processamento de dados é importante pois é nela que informações acerca do comportamento da cidade serão determinadas. Os dados processados, neste caso, podem apontar áreas que mais registram ocorrências, bem como qual o tipo mais frequente. Uma área, por exemplo, pode ser marcada por enchentes ou alagamento quando chove. Portanto, tal característica pode ser detectada.

Ocorrências reportadas à plataforma de SP podem detectar problemas. Alguns exemplos são listados a seguir.

- Áreas com mais alagamentos;
- Áreas menos seguras;
- Rotas mais utilizadas;
- Destinos frequentes de acordo com o trajeto efetuado;
- Estado do pavimento;
- Principal problema registrado em determinada rua.

Com estatísticas e conclusões deste tipo, é possível reconhecer problemas com mais rapidez e precisão. Portanto, o sistema pode auxiliar na tomada de decisão por parte dos governantes.

4.3.1 Mapa Auto-Organizável

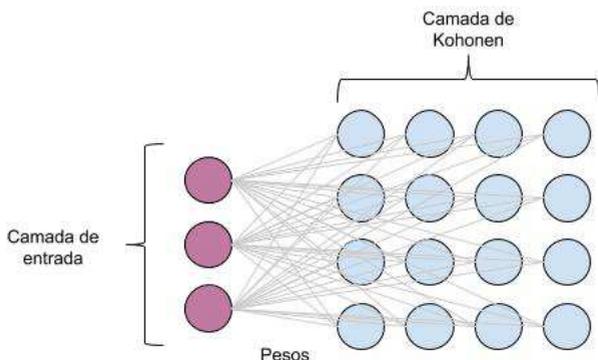
Mapa Auto-Organizável¹² (SOM) é um tipo de Rede Neural Artificial (RNA) de aprendizado não supervisionado desenvolvida por Teuvo Kohonen.

A fim de obter informações com contexto e uma visão integrada dos dados enviados pelos usuários, é realizado um processamento de dados utilizando Mapa Auto-Organizável. Com ele, é possível visualizar de forma combinatória a expressividade de cada atributo do conjunto de dados. Ou seja, com o mapa de componentes é possível visualizar ocorrências registradas em um região que podem ser consideradas situações negativas que devem ser amenizadas. Além disso, pode-se fazer uma interpretação da influência dos *clusters* e seus vizinhos através da *U-Matrix* de acordo com a coloração dos *clusters* gerados.

¹²Tradução de *Self-Organizing Map*

O funcionamento do SOM consiste em agrupar as entradas similares em *clusters*. O SOM é composto pelas camadas de entrada e oculta, também chamada de camada de Kohonen. A primeira recebe a entrada dos dados e a segunda é a que possui os neurônios que serão treinados e mostrarão a representação em duas dimensões (KOHONEN, 2000). A Figura 12 ilustra um exemplo de rede SOM com três neurônios de entrada e uma camada de Kohonen 4×4 .

Figura 12 – Mapa Auto-Organizável.



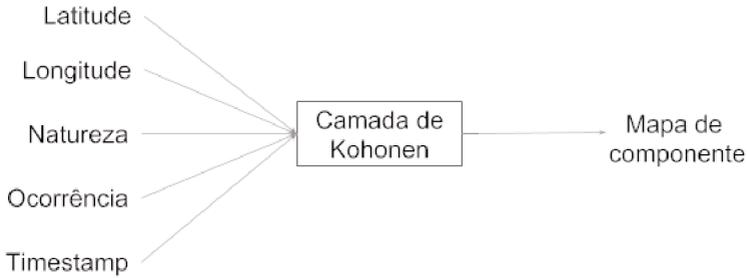
Fonte: Elaborado pelo autor.

Para cada entrada, é verificado qual é o neurônio de menor distância da camada de Kohonen, chamado de *Best Matching Unit* (BMU). Os neurônios mais próximos do BMU serão mais influenciados pelo ajuste dos pesos em cada conjunto de dados da camada de entrada. Ou seja, dado um conjunto de dados x , para cada conjunto de entrada x_i , o BMU será escolhido e os neurônios mais próximos dele serão mais influenciados no ajuste dos pesos. Esse processo se repete na próxima iteração, com o conjunto de entrada x_{i+1} . Ao final do treinamento, os neurônios estarão organizados em *clusters*, resultando numa representação em duas dimensões dos dados de entrada (KOHONEN, 2000).

Os registros de ocorrências realizados pelos usuários através do aplicativo possuem os seguintes atributos: Localização (Latitude e Longitude), Natureza, Ocorrência e *Timestamp*. Portanto, a fim de visualizar a expressividade de cada um deles, gera-se o mapa de componentes, em que os dados de entrada são os atributos das ocorrências registradas

e a saída é o mapa de componente.

Figura 13 Dados de entrada e saída do SOM.



Fonte: Elaborado pelo autor.

O SOM realiza um treinamento com cada dado de entrada para, então, gerar o mapa de componente. Os pesos de cada entrada mostrados na Figura 13 representam o vetor de pesos para cada neurônio da rede.

4.3.2 Acesso à Informação

Conforme já mencionado, a coleta dos dados através dos *smartphones* é realizada com o intuito de entender a dinâmica da cidade em questão.

Nesta arquitetura, qualquer pessoa pode acessar os dados. Além disso, o objetivo é que os líderes políticos interajam com os dados processados, considerando a perspectiva dos cidadãos e a demanda da população na solução de um problema, ou seja, que haja uma alocação de recursos eficiente. Os usuários também podem acessar seu histórico de ocorrências e de dados enviados. Por exemplo, acessar um mapa de ocorrências.

5 APLICAÇÃO DESENVOLVIDA

Neste capítulo, a aplicação Android é apresentada, bem como suas etapas de desenvolvimento. O funcionamento da aplicação, da plataforma InterSCity e as ferramentas e tecnologias utilizadas são retratados em detalhes.

5.1 IMPLEMENTAÇÃO

A arquitetura apresentada no Capítulo 4 consiste, basicamente, em duas partes essenciais: aplicação e *middleware*. A comunicação entre elas ocorre por meio de mensagens via protocolo HTTP. Este assunto é tratado detalhadamente na Seção 5.4. Para fins de estudo da viabilidade desta arquitetura, uma aplicação Android para Sensoriamento Participativo, denominada InterageAru, foi desenvolvida e o InterSCity foi o *middleware* utilizado.

A aplicação oferece suporte às funções básicas de um aplicativo voltado para SP de uma cidade. É possível efetuar cadastro de cidadãos que ainda não são usuários do sistema.

Visando atingir a maior parte do mercado de *smartphones* do Brasil, o aplicativo foi desenvolvido para a plataforma Android. Este é um sistema operacional móvel desenvolvido pela Google. Criado pela mesma empresa, o ambiente de desenvolvimento integrado¹³ (IDE) utilizado para o desenvolvimento desta aplicação foi o Android Studio, versão 3.1.1, juntamente com o *Java Development Kit* (JDK) versão 8.

5.2 CONFIGURAÇÃO DO INTERSCITY

A plataforma InterSCity possui algumas dependências de funcionamento. Como os seus microsserviços tem funcionamento independente, a tecnologia Docker foi utilizada para execução de cada um desses. Cada microsserviço é considerado uma aplicação multicontainer, pois envolve containers de propósitos mais específicos. Por isso, a ferramenta Docker Compose também foi utilizada na configuração do InterSCity. Por exemplo, o Código 1 mostra o trecho extraído na inicialização dos containers do microsserviço `Data Collector`.

¹³Tradução de *Integrated Development Environment*

 Código 1 – Containers do microsserviço Data Collector

```
Starting data-collector :
Starting data-collector-mongo    ... done
Starting data-collector-mongo-cache ... done
Starting data-collector          ... done
```

Após a configuração das dependências e da plataforma, o acesso a cada microsserviço é realizado através de uma porta. A porta de cada serviço é listada na Tabela 3.

Tabela 3 – Porta dos microsserviços.

Microsserviço	Porta
Resource Adaptor	3002
Resource Catalog	3000
Actuator Controller	5000
Data Collector	4000
Resource Discovery	3004

Fonte: Elaborado pelo autor.

Para esta aplicação, somente alguns serviços foram utilizados, estes são: **Resource Adaptor**, **Resource Catalog** e **Data Collector**. Dependendo do contexto e do método de solicitação HTTP, a comunicação foi estabelecida através de diferentes *urls*. Uma explicação mais detalhada desta etapa é descrita na Seção 5.4.

5.3 INTERAGEARU

Como Araranguá é a cidade escolhida como objeto de estudo deste cenário, o aplicativo recebeu o nome de InterageAru. Porém, este é um mero detalhe, visto que o aplicativo pode ser replicado em qualquer cidade, pois a implementação do InterageAru não dependeu das características intrínsecas de Araranguá.

Nas próximas seções, serão descritas detalhadamente as principais configurações da aplicação.

5.3.1 Activities

Uma das classes mais importantes no Android é a *Activity*. Basicamente esta classe é uma representação de uma tela na aplicação. É nela que serão definidas as funcionalidades e a interface gráfica. A Figura 14 exibe as interfaces do aplicativo InterageAru.

Figura 14 Interfaces do aplicativo InterageAru.

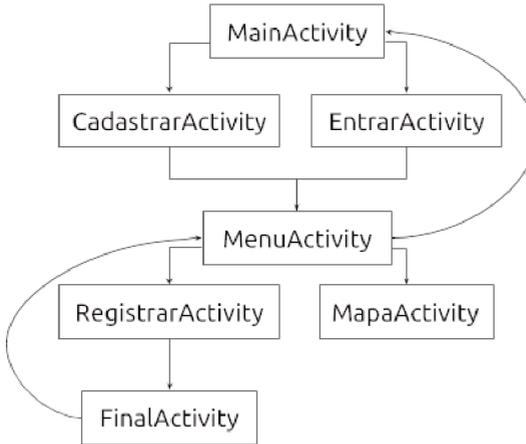


Fonte: Elaborado pelo autor.

O InterageAru é composto por sete interfaces: *MainActivity*, *CadastrarActivity*, *EntrarActivity*, *MenuActivity*, *RegistrarActivity*, *MapaActivity* e *FinalActivity*. A Figura 15 ilustra o fluxo

de dados entre estas interfaces.

Figura 15 Fluxo de telas.



Fonte: Elaborado pelo autor.

Para melhor entendimento da Figura 15, as funcionalidades de cada interface são descritas resumidamente a seguir:

- **MainActivity**: interface inicial do aplicativo. Dá acesso às interfaces de cadastro e de *login*;
- **CadastrarActivity**: permite que o usuário se cadastre na aplicação;
- **EntrarActivity**: nega ou concede o acesso à aplicação após o usuário informar seu CPF;
- **MenuActivity**: dá acesso às principais funcionalidades do aplicativo;
- **RegistrarActivity**: permite que o usuário selecione a natureza, o tipo de ocorrência (conforme a Figura 8) e envie o registro com a localização e o *timestamp* para o *middleware*;
- **MapaActivity**: possibilita a visualização dos dados no mapa (não implementado na versão protótipo);

- **FinalActivity**: exibe confirmação do envio dos dados e retorna ao menu principal.

5.4 COMUNICAÇÃO COM INTERSCITY

A comunicação entre a aplicação Android e a plataforma InterSCity é crucial neste cenário. Sem ela, pode haver perda de registro de ocorrências e não há acesso aos dados da plataforma.

Dependendo da função do aplicativo utilizado, a comunicação é estabelecida em diferentes `urls`. As principais funcionalidades do InterageAru e os microsserviços do InterSCity utilizados por esta aplicação com suas respectivas `urls` estão organizados na Tabela 4.

Tabela 4 – Método de solicitação por funcionalidade

Funcionalidade	Método	url
Cadastrar <i>capability</i>	POST	IP:3000/ <i>capabilities</i>
Cadastrar recurso	POST	IP:3002/ <i>resources</i>
<i>Login</i>	GET	IP:3000/ <i>resources/search?description=cpf</i>
Registrar ocorrência	POST	IP:3002/ <i>resources/uuid/data/participatory_sensing</i>
Consultar todas as ocorrências	POST	IP:4000/ <i>resources/data</i>

Fonte: Elaborado pelo autor.

Conforme mencionado no Capítulo 4, a troca de mensagens entre o InterageAru e o InterSCity acontece via protocolo HTTP, em que os dados são trocados em formato JSON.

O tratamento dos dados enviados e recebidos variam de acordo com o contexto. No InterSCity, os recursos devem enviar dados referentes à uma *capability* cadastrada na plataforma. Caso contrário, o dado não será armazenado. Uma *capability* é uma categorização de um tipo de medição, ou de dado enviado. O modelo JSON para cadastrar uma *capability* deve estar estruturado da forma apresentada no Código 2. Devem ser especificados o nome, uma descrição e o tipo associado à essa *capability*.

 Código 2 – Modelo JSON de cadastro de uma nova capability

```
{
  "name": "participatory_sensing",
  "description": "Send information about the city",
  "capability_type": "sensor"
}
```

Os recursos, assim como as *capabilities*, devem ser cadastrados seguindo um modelo JSON; um exemplo do trecho de código é mostrado no Código 3. A descrição, neste caso, é o CPF informado pelo usuário através do aplicativo InterageAru; todos os recursos são associados à *capability* `participatory_sensing` e tem `status` `active`; a latitude e a longitude são as atuais do *smartphone*.

 Código 3 – Modelo JSON de cadastro de um recurso

```
{
  "data": {
    "description": "78421134400",
    "capabilities": [
      "participatory_sensing"
    ],
    "status": "active",
    "lat": -28.9413753,
    "lon": -49.4906649
  }
}
```

As ocorrências registradas pelos usuários através do aplicativo InteraguAru são vistas pela plataforma InterSCity como os dados enviados pelos recursos. Conforme descrito no Capítulo 4, os dados enviados são as ocorrências registradas pelos usuários em determinado local da cidade. Portanto, o modelo JSON de registro de ocorrência deve informar a localização, data e hora, a ocorrência e sua respectiva natureza. O Código 4 ilustra um exemplo de cadastro de uma ocorrência. Conforme mostrado, esta ocorrência é “Buraco no pavimento”, de natureza “Infraestrutura”, registrada no dia 3 de maio de 2018 às 17h44 GMT na localização indicada.

 Código 4 – Modelo JSON de registro de ocorrência

```
{
  "data": {
    "location": {
```

```

        "lat": -28.938887,
        "lon": -49.489600
    },
    "natureza": "Infraestrutura",
    "ocorrencia": "Buraco no pavimento",
    "timestamp": "2018-05-03T17:44:47.042Z"
}
}
}

```

5.5 APLICATIVO EM FUNCIONAMENTO

Nesta Seção serão demonstradas as funcionalidades do aplicativo. As respostas das solicitações HTTP serão conferidas através de informações coletadas com a API Postman e do LOG do Android Studio.

A fim de o usuário ter acesso à tela de registro de ocorrência, é necessário que ele tenha realizado o cadastro na plataforma. Este ocorre quando o usuário acessa a `CadastrarActivity`, informa o CPF e aperta o botão `CADASTRAR` (Figura 16). A Figura 32 exibe as informações deste cadastro na plataforma InterSCity. É possível observar que a descrição é o CPF informado e o `uuid` atribuído confere com o recebido pelo aplicativo, exibido no LOG do Android Studio (Figura 18).



Figura 16 Cadastro de usuário.



Figura 17 Usuário cadastrado na plataforma.

Figura 18 LOG exhibe uuid atribuído.

```
06-07 13:19:58.787 26619-26619/mrdaros.com.github.interagearu I/TAG: UUID Menu: 9004ad97-b55a-4e08-b5c8-5612f99b80e3
06-07 13:19:58.928 26619-26619/mrdaros.com.github.interagearu I/TAG: UUID: 9004ad97-b55a-4e08-b5c8-5612f99b80e3
```

Fonte: Elaborado pelo autor.

Quando o usuário já efetuou seu cadastro, o acesso é feito através da `EntrarActivity`. Ela é similar à interface de cadastro. O usuário informa o CPF e aperta o botão **ENTRAR** (Figura 19). Pode-se observar que o `uuid` recebido pelo aplicativo após a solicitação (Figura 21) confere com o registrado na plataforma (Figura 20).



Figura 19 Entrar com CPF já cadastrado.

```
1 = k
2 -
3 -
4   "resources": [
5     {
6       "id": 44361,
7       "uri": null,
8       "created_at": "2018-05-09T12:26:26.845Z",
9       "updated_at": "2018-05-09T12:26:28.022Z",
10      "lat": -28.9667314317533,
11      "lon": -49.5371751710259,
12      "status": "active",
13      "collect_interval": null,
14      "description": "86859667796",
15      "uuid": "b63c2781-7a12-401e-8ff0-0cd35ee3c754",
16      "city": "Araucária",
17      "neighborhood": null,
18      "state": "Santa Catarina",
19      "postal_code": "88900-000",
20      "country": "Brazil",
21      "capabilities": [
22        "participatory_sensing"
23      ]
24    }
25  ]
26 }
```

Figura 20 Informações registradas na plataforma.

Figura 21 LOG exhibe uuid após busca.

```
06-07 13:14:29.755 26619-26619/mrdaros.com.github.interagearu I/EntrarActivity: uuid search: b63c2781-7a12-401e-8ff0-0cd35ee3c754
06-07 13:14:30.002 26619-26619/mrdaros.com.github.interagearu I/TAG: UUID Menu: b63c2781-7a12-401e-8ff0-0cd35ee3c754
```

Fonte: Elaborado pelo autor.

Se o usuário tentar entrar com um CPF não cadastrado o acesso não é concedido. A Figura 22 ilustra a mensagem exibida durante uma tentativa de acesso. Pode-se perceber que, como não há nenhum registro com este CPF, não há retorno de nenhum `uuid` nesta tentativa, conforme exibido no LOG (Figura 23).

Figura 22 Mensagem informa que não está cadastrado.



Fonte: Elaborado pelo autor.

Figura 23 LOG com uuid não retornado.

```
06-07 13:09:16.406 26619-26619/mrdaros.com.github.interagearu I/EntrarActivity: uuid search:
```

Fonte: Elaborado pelo autor.

A última funcionalidade do aplicativo que depende da plataforma InterSCity é o registro de ocorrência. Na `RegistrarActivity` é possível selecionar um dos três tipos de Natureza e uma de suas respectivas ocorrências ao apertar as setas. A latitude e longitude são exibidas automaticamente. A Figura 24 ilustra o exemplo executado. Pode-se observar que o registro foi efetuado na plataforma ao conferir as informações exibidas na Figura 25. Exibidos na Figura 26, o `uuid` registrado no LOG confere e a `url` segue com a conformidade adotada na Tabela 4.

The screenshot shows a web form titled "InterageAru". It has four input fields: "Natureza" with a dropdown menu set to "Infraestrutura", "Ocorrência" with a dropdown menu set to "Queda de energia", "Latitude" with the value "-28.948675", and "Longitude" with the value "-49.4979317". At the bottom right of the form is a button labeled "ENVIAR".

Figura 24 Envio de ocorrência.

```
{
  "resources": [
    {
      "uuid": "9804ad97-b55a-4e08-b5c8-5612f99b80e3",
      "capabilities": {
        "participatory_sensing": [
          {
            "location": {
              "lat": -28.948675,
              "lon": -49.4979317
            },
            "natureza": "Infraestrutura",
            "ocorrencia": "Queda de energia",
            "date": "2018-06-07T16:23:10.637Z"
          }
        ]
      }
    }
  ]
}
```

Figura 25 Ocorrência cadastrada na plataforma.

Figura 26 LOG exibe uuid e url.

```
06-07 13:19:58.928 26619-26619/mrdaros.com.github.interagearu I/TAG: UUID: 9804ad97-b55a-4e08-b5c8-5612f99b80e3
06-07 13:21:04.559 26619-26619/mrdaros.com.github.interagearu I/TAG: UUID Registrar: 9804ad97-b55a-4e08-b5c8-5612f99b80e3
URL Registrar: http://190.162.234.26:3002/resources/9804ad97-b55a-4e08-b5c8-5612f99b80e3/data/participatory\_sensing
```

Fonte: Elaborado pelo autor.

6 AVALIAÇÃO DO SISTEMA PROPOSTO

Neste capítulo são apresentados os procedimentos realizados para avaliar o funcionamento da plataforma InterSCity, do aplicativo InterageAru e dos resultados obtidos após o processamentos dos dados executado pelo Mapa Auto-Organizável de Kohonen.

6.1 PROCEDIMENTOS ADOTADOS

Para avaliar as duas principais partes pertencentes à arquitetura - o aplicativo InterageAru e a plataforma InterSCity -, foram realizados testes que enviam ocorrências ao InterSCity. O primeiro teste realizado enviou dados à plataforma através do aplicativo InterageAru. Nestes testes, foi verificado se os dados enviados foram registrados na plataforma e se a comunicação entre o InterageAru e o InterSCity funciona corretamente. Além disso, o desempenho do aplicativo foi comparado com o tempo de resposta das solicitações HTTP com o auxílio da API Postman.

A segunda parte de avaliação da arquitetura consistiu em verificar se é possível obter informações inteligentes após o processamento de dados utilizando SOM. Neste tipo de avaliação, primeiro foram processados dados com uma predominância da ocorrência “Buraco no pavimento”. Também foi realizado um último teste com dados aleatórios de mesma probabilidade. Em cada um desses três testes, foram obtidos os mapas de cada atributo - localização, natureza, ocorrência e *timestamp* - e os *clusters* com a média de cada um desses atributos. Sobre os dados enviados para a plataforma, também foi gerado relatório de frequência de ocorrência. Esta etapa foi realizada utilizando a linguagem de programação Python, versão 2.7.15, e seus pacotes.

6.2 AVALIAÇÃO DO APLICATIVO

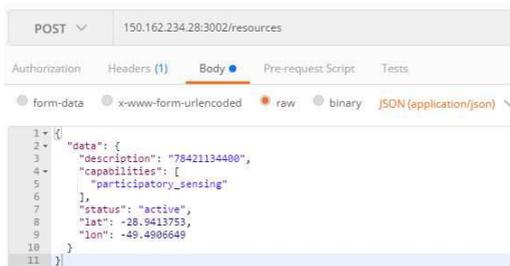
Nesta Seção, faz-se uma avaliação do desempenho do aplicativo por comparações entre o tempo de resposta das solicitações HTTP com a API Postman e com o InterageAru.

O tempo de resposta de cada solicitação HTTP foi medido, primeiramente, através da API Postman, a fim de compará-lo, posteriormente, com o obtido pelo aplicativo InterageAru. Os resultados foram

obtidos durante o teste descrito na Seção 5.5. As mensagens de tempo são exibidas logo após chamada da *Activity* seguinte. Este teste seguiu os métodos de solicitação por funcionalidade listados na Tabela 4.

A simulação de cadastro foi a primeira solicitação feita com o Postman. A Figura 27 mostra o modelo JSON do recurso cadastrado; ele tem como descrição um CPF válido e tem a *capability participatory_sensing* atribuída a ele.

Figura 27 – Simulação de cadastro de recurso com Postman.



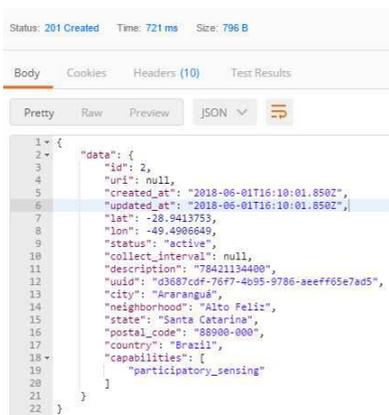
```

1 {
2   "data": {
3     "description": "78421134400",
4     "capabilities": [
5       "participatory_sensing"
6     ],
7     "status": "active",
8     "lat": -28.9413753,
9     "lon": -49.4906649
10  }
11 }

```

Fonte: Elaborado pelo autor.

Figura 28 – Retorno da simulação de cadastro de recurso.



```

Status: 201 Created Time: 721 ms Size: 796 B
Body Cookies Headers (10) Test Results
Pretty Raw Preview JSON
1 {
2   "data": {
3     "id": 2,
4     "url": null,
5     "created_at": "2018-06-01T16:10:01.850Z",
6     "updated_at": "2018-06-01T16:10:01.850Z",
7     "lat": -28.9413753,
8     "lon": -49.4906649,
9     "status": "active",
10    "collect_interval": null,
11    "description": "78421134400",
12    "uid": "d3687cdf-76f7-4b95-9786-aeef65e7ad5",
13    "city": "Aranangus",
14    "neighborhood": "Alto Feliz",
15    "state": "Santa Catarina",
16    "postal_code": "88900-000",
17    "country": "Brazil",
18    "capabilities": [
19      "participatory_sensing"
20    ]
21  }
22 }

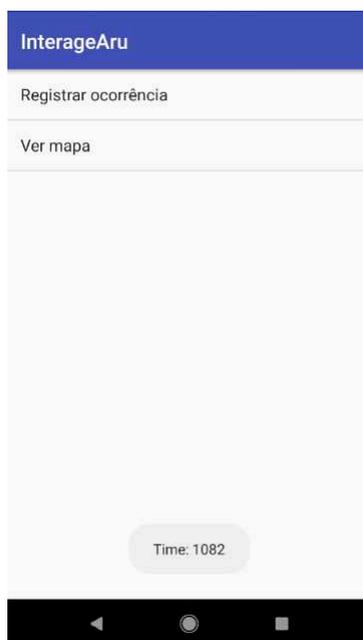
```

Fonte: Elaborado pelo autor.

O retorno da solicitação pode ser visto na Figura 28. Observa-se que o tempo de resposta é de 721 ms.

Efetuada o cadastro com o aplicativo InterageAru, o tempo de resposta é de 1082 ms, conforme mostrado na Figura 29. Ponderando que este tempo engloba a geração do JSON e a chamada da próxima *Activity*, ele é considerado aceitável, pois é somente 361 ms mais lento que o da simulação.

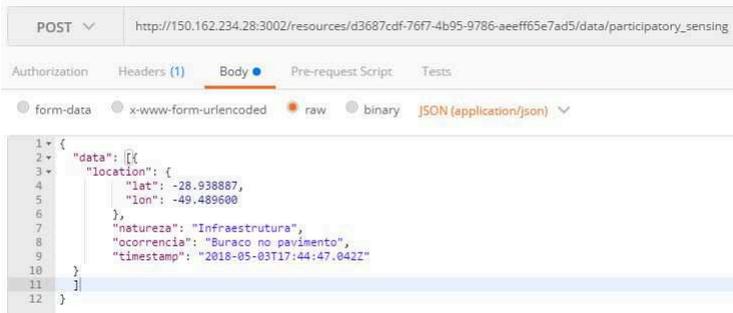
Figura 29 – Tempo de resposta de cadastro de recurso com o aplicativo.



Fonte: Elaborado pelo autor.

A segunda simulação realizada foi a de envio de ocorrência para a plataforma. Neste caso, o dado enviado está na latitude -28.938887 e longitude: -49.489600; a natureza é "Infraestrutura", ocorrência "Buraco no pavimento" e foi registrada no dia 3 maio 2018, às 17h44.

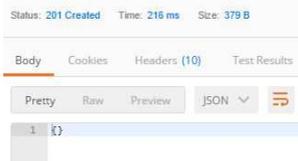
Figura 30 – Simulação de envio de ocorrência com Postman.



Fonte: Elaborado pelo autor.

O retorno da solicitação pode ser visto na Figura 31. Ela não possui dados de retorno, mas observa-se que o tempo de resposta é de 216 ms.

Figura 31 – Retorno de envio de ocorrência com Postman.



Fonte: Elaborado pelo autor.

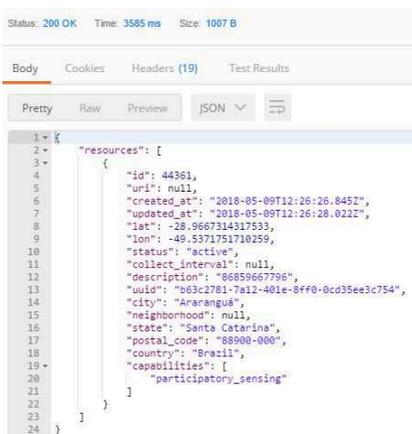
O tempo de resposta, após o envio de ocorrência pelo aplicativo InterageAru, ilustrado na Figura 32, e é de 704 ms. Comparando com o resultado obtido com o Postman, exibido anteriormente, pode-se concluir que esta diferença de aproximadamente 500 ms é aceitável, pois também envolve a geração do JSON e a chamada da próxima *Activity*. Além disso, é pouco perceptível ao usuário e não interfere no funcionamento do aplicativo.

Figura 32 – Tempo de resposta de envio de ocorrência com InterageAru.



Fonte: Elaborado pelo autor.

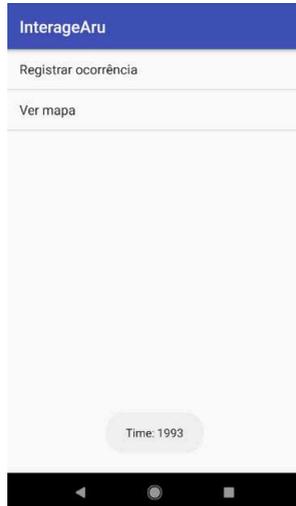
Figura 33 – Busca de recurso com Postman.



Fonte: Elaborado pelo autor.

E, por último, foi medido o tempo de resposta de busca de recurso pela descrição. Neste caso, através do método GET, a solicitação feita para a URL `http://150.162.234.28:3000/resources/search?description=86859667796` e o retorno obtido pela API é ilustrado na Figura 33.

Figura 34 – Tempo de resposta com o aplicativo.



Fonte: Elaborado pelo autor.

Percebe-se que o tempo de resposta de 3585 ms aumenta consideravelmente em relação às outras solicitações, pois é necessário encontrar o recurso com a descrição correspondente.

Entretanto, pelo aplicativo, o tempo de resposta diminuiu consideravelmente para 1993 ms. Isto pode ter sido causado pela qualidade da conexão. Apesar disso, este tempo ainda é considerado ligeiramente alto. Uma possível solução é utilizar a API `SharedPreferences` para salvar a sessão do usuário e evitar a necessidade de efetuar o *login*.

6.3 AVALIAÇÃO DA FASE DE PROCESSAMENTO DOS DADOS

Nesta etapa de avaliação foi necessário enviar uma grande quantidade de dados para a plataforma, a fim de classificá-los com algum

algoritmo de classificação de dados.

Um algoritmo de geração de dados foi criado para enviar os dados e o pacote `sompy` foi utilizado para poder inferir informações acerca dos dados, com o auxílio do mapa de Kohonen, *U-matrix* e *k-means*.

6.3.1 Geração de Dados

Para o algoritmo SOM poder obter resultados mais conclusivos e gerar *clusters* que representem uma quantidade significativa de dados que tenham atributos similares, é necessário trabalhar com algumas centenas de dados. Por isso, nesta etapa, foi criado um algoritmo de geração de recursos e dados para serem enviados para a plataforma.

Este algoritmo também foi utilizado como ferramenta para verificar a taxa de sucesso de envio de dados e cadastro de recursos na plataforma, apresentados na Seção 6.4.

Para o teste com dados aleatórios, ou seja, sem predominância de nenhuma ocorrência, este algoritmo seguiu as premissas de registro e de ocorrências apresentados no Capítulo 4 e são descritas a seguir:

- A descrição do recurso é o CPF do usuário, portanto um CPF válido é gerado;
- As ocorrências válidas são os tipos mostrados anteriormente na Figura 8;
- A localização está limitada à cidade de Araranguá (SC)
 - As máxima e mínima latitudes são -28.912090 e -28.999855 , respectivamente.
 - As máxima e mínima longitudes são -49.436552 e -49.556962 , respectivamente.
- A data segue o padrão ISO 8601 com registro mínimo de 1 fev. 2018 às 12:00 e máximo de 30 abr. 2018 às 12:00.

No entanto, nos testes com somente parte dos dados gerados aleatoriamente, os atributos alterados são mostrados nas tabelas a seguir. A Tabela 5 lista o conjunto de dados do primeiro teste realizado e possui 60% das ocorrências como “Buraco no pavimento” nas latitudes e longitudes estipuladas.

Tabela 5 – Atributos do primeiro conjunto de dados.

Teste	Ocorrência	%	Latitude	Longitude
1	Buraco no pavimento	30%	-28.949534	-49.486293
		30%	-28.937335	-49.484345

Fonte: Elaborado pelo autor.

Primeiramente, o algoritmo, através da função `createResource()`, cria um modelo JSON para um novo recurso com um CPF válido e uma latitude e longitude dentro dos limites estabelecidos. Depois, estabelece uma conexão com o InterSCity e o registra. O `uuid` é extraído da resposta da solicitação HTTP e este é utilizado para estabelecer uma segunda conexão com o InterSCity. Mas, dessa vez, para envio de dados.

A função `createData()` cria um modelo JSON de um dado novo. Assim como na função `createResource()`, a latitude e longitude pertencem a um intervalo. Uma natureza e sua respectiva ocorrência são selecionadas e uma data e hora são geradas. Com o modelo criado, ele é enviado após estabelecer uma conexão com a plataforma.

6.3.2 Tratamento dos Dados

Os dados utilizados nesta etapa foram os mesmos enviados pelo algoritmo descrito na Seção 6.3.1. Foram obtidos através de uma solicitação HTTP, conforme Tabela 4, em formato JSON. Para poder realizar o processamento dos dados, eles foram escritos em formato `.csv` e, em seguida, convertidos em um *Data Frame* do pacote `pandas`.

Em Mapa Auto-Organizável, é necessário normalizar os dados que não estejam em formatos numéricos, pois é preciso calcular a distância entre os neurônios e encontrar o BMU. Portanto, os dados natureza, ocorrência e *timestamp* precisaram ser normalizados. Como as naturezas e as ocorrências são armazenadas como cadeia de caracteres na plataforma, o método adotado foi a conversão para índices quando os dados foram escritos no arquivo `.csv` (Tabela 6). O *timestamp*, entretanto, convertido para o formato `datetime`, é normalizado entre $[0, 1]$ com a função `MinMaxScaler` do pacote `sklearn`.

Tabela 6 – Índices por natureza e ocorrência

	Tipo	Índice
Natureza	Infraestrutura	1
	Mobilidade	2
	Segurança	3
Infraestrutura	Buraco no pavimento	1
	Calçada sem manutenção	2
	Desmoronamento	3
	Poste caído	4
	Iluminação deficiente	5
	Vazamento de água	6
	Enchente	7
	Queda de energia	8
	Sinalização de trânsito	9
Mobilidade	Acidente de trânsito	10
	Obra	11
	Congestionamento	12
Segurança	Assalto	13
	Furto	14

Fonte: Elaborado pelo autor.

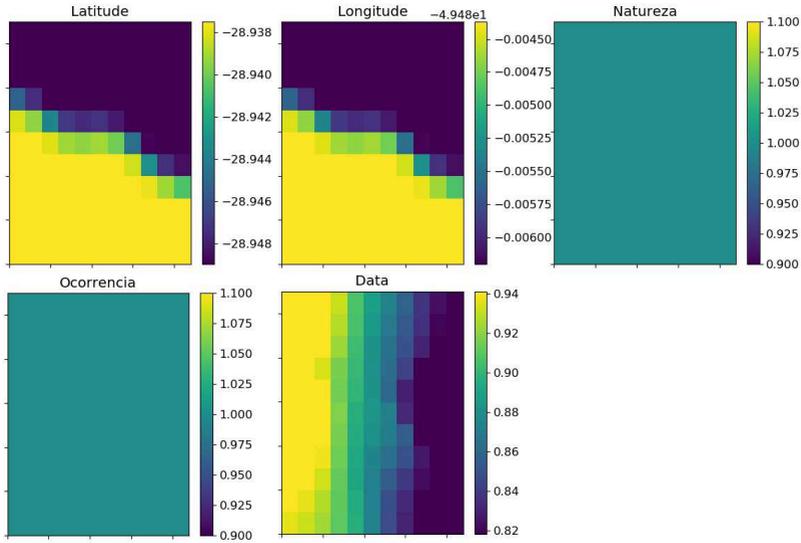
Após o tratamento dos dados, os três testes foram realizados. O processamento de dados utilizando SOM foi feito com o auxílio do pacote `sompy`. O tamanho da camada de Kohonen é calculado pela biblioteca de acordo com os valores dos dados.

6.3.3 Teste de ocorrência

Este teste foi executado sobre as amostras de dados enviadas de acordo com a Tabela 5. Primeiramente, a análise é feita sobre os dados de controle, os 60% com ocorrência e localização conhecidas. Em seguida, sobre todo o conjunto de dados, com os 40% restante aleatório.

O primeiro resultado exibido pela biblioteca é o mapa de componentes. Nele, é mostrada a expressividade de cada componente. Ressalta-se que o mapa de neurônios é o mesmo para cada componente, ou seja, os neurônios centrais do mapa de Latitude são os mesmos para Longitude e os demais.

Figura 35 – Mapa de componentes do primeiro teste de ocorrência.

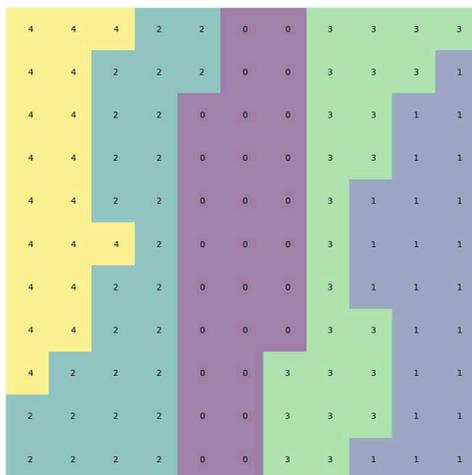


Fonte: Elaborado pelo autor.

A partir dos mapas ilustrados na Figura 35 pode-se concluir que a natureza e a ocorrência observadas são, respectivamente, “Infraestrutura” e “Buraco no pavimento”, conforme esperado. Percebe-se que há predominância das Latitudes -28.9495 (azul) e -28.9373 (amarelo) e das Longitudes -49.486293 (azul) e -49.484345 (amarelo). A latitude em azul corresponde à longitude em azul, conforme descrito na Tabela 5. O mesmo acontece para a amarela.

Da Figura 35 pode-se concluir que em todas as localizações possuem registro de natureza e ocorrência com índice 1 e que os registros predominaram no fim do período, pois a data varia de 0.8 a 0.95.

Figura 36 – *Clusters* do primeiro teste de ocorrência.



Fonte: Elaborado pelo autor.

Código 5 – Lista de clusters do primeiro teste de ocorrência

Topographic error = 0.18666666666666668; Quantization error = 0.0375169942530162

Performing K-means SSE elbow sweep...

Centroid 0[-28.94302387 -49.48525343 1. 1. 0.88146154]

Centroid 1[-28.94453029 -49.48549398 1. 1. 0.7921078]

Centroid 2[-28.94203343 -49.48509527 1. 1. 0.92744066]

Centroid 3[-28.94378874 -49.48537557 1. 1. 0.8360292]

Centroid 4[-28.94295521 -49.48524246 1. 1. 0.97001219]

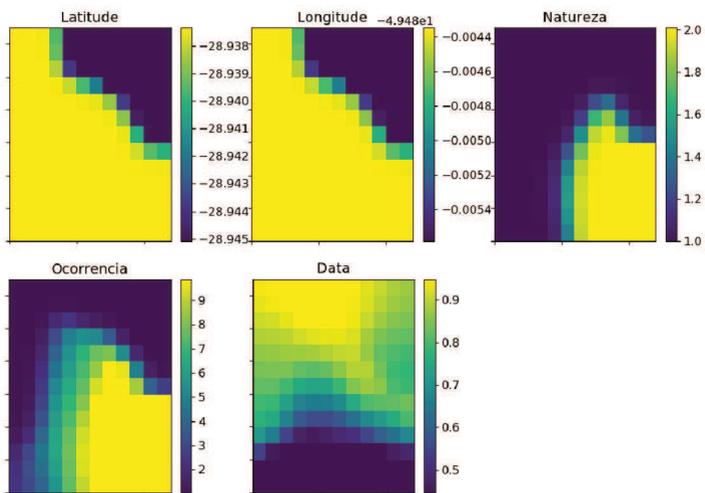
Sobre o mapa de Kohonen obtido, a biblioteca executa o método *k-means* e calcula os centróides dos *clusters* dos dados, mostrados no Código 5. Através dessa lista é possível fazer uma previsão de que tipo de ocorrência será registrada em um local. Por exemplo, dada a localização $(-28.9445, -49.4854)$, pode-se dizer que a mais provável ocorrência neste local será “Buraco no pavimento” com data próxima ao início da última semana do período registrado.

Os centróides são importantes para gerar informações acerca de uma região da cidade, a fim de entender o tipo de ocorrência mais recorrente.

Executando a biblioteca sobre todo o conjunto de dados, percebe-se que os resultados são uma extensão dos anteriores. Observando o mapa de componentes na Figura 37, conclui-se que, além do que já foi observado, é registrado no local $(-28.937335, -49.484345)$ outras ocorrências da natureza “Infraestrutura” e “Mobilidade”. Como o número de ocorrências da natureza “Segurança” é muito menor que o das outras, ele não aparece no mapa, devido à intensidade. Conforme o esperado, pode-se ainda observar que:

- Na localização $(-28.945, -49.4862)$ há registro de Infraestrutura, ocorrência Buraco no pavimento no fim do período observado, mais precisamente na última semana.
- A localização $(-28.9373, -49.4843)$ registra aproximadamente 50% de registro Infraestrutura, ocorrência Buraco no pavimento também no fim do período observado.
- A localização $(-28.9373, -49.4843)$ registra aproximadamente 30% de registro Mobilidade, ocorrência Acidente de trânsito na metade do período observado.
- São registradas ocorrências de Sinalização de trânsito (9), Queda de energia (8), Enchente (7), Vazamento de água(6), Iluminação deficiente (5), Poste caído (4), Desmoronamento (3) e Calçada sem pavimento (2) na localização $(-28.9373, -49.4843)$, distribuídos entre a metade do período até o início da última semana.

Figura 37 Mapa de componentes do segundo teste de ocorrência.



Fonte: Elaborado pelo autor.

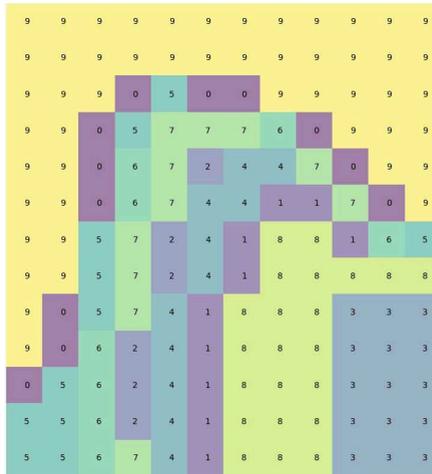
Figura 38 *U-matrix* do segundo teste de ocorrência.



Fonte: Elaborado pelo autor.

A segunda observação a ser feita neste teste é a *U-matrix* gerada pela biblioteca. Ela auxilia a observar os neurônios do mapa de componentes em *clusters*. Visualizando as Figuras 38 e 39, percebe-se que os neurônios com coloração similar são organizados em um *cluster*.

Figura 39 – *Clusters* do segundo teste de ocorrência.



Fonte: Elaborado pelo autor.

Observando a Figura 39, percebe-se que os *clusters* são uma combinação dos mapas de componente. E a ocorrência mais registrada de cada região é mostrada de acordo com os centróides gerados, mostrados no Código 6.

Código 6 – Lista de clusters do segundo teste de ocorrência

```

Topographic error = 0.013; Quantization error =
0.10000143617447355
Performing K-means SSE elbow sweep...
Centroid 0[-28.94207598 -49.48510206 1.01236732 1.70231783
0.82822413]
Centroid 1[-28.93777163 -49.48441472 1.6125464 9.32831477
0.60353088]
Centroid 2[-28.93735339 -49.48434794 1.01065684 6.47157451
0.55805554]
Centroid 3[-28.93735408 -49.48434805 2.87951338 13.22712779

```

```

0.46145046]
Centroid 4[-28.93747709 -49.48436769 1.11296462 7.7685931
0.6080833 ]
Centroid 5[-28.93889502 -49.48459411 1.02362449 2.77344145
0.59202034]
Centroid 6[-28.93945994 -49.48468432 1.04850976 4.03740692
0.59621321]
Centroid 7[-28.93928405 -49.48465624 1.06920211 5.39919967
0.7600745 ]
Centroid 8[-28.93759545 -49.48438659 2.06679699 10.97166237
0.53775518]
Centroid 9[-28.94390294 -49.4853938 1.00056254 1.04298981
0.89010914]

```

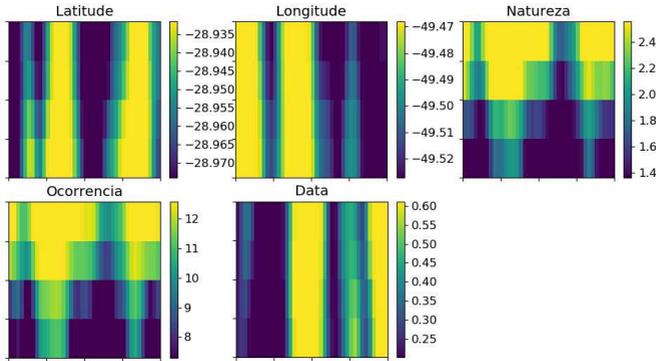
6.3.4 Teste de Aleatoriedade

Este teste utilizou um conjunto de dados aleatório para todos os atributos de envio de ocorrência.

Observando o mapa de componentes na Figura 37, conclui-se que, a maioria das naturezas registradas é do tipo Infraestrutura e Segurança. Devido à intensidade das ocorrências de índices entre 7 e 13, as restantes não são exibidas. As seguintes observações mais pontuais podem ser feitas a partir dos mapas de componentes:

- Na localização $(-28.979, -49.515)$ há uma maioria de registros de Infraestrutura, ocorrência Enchente(7), Queda de Energia(8) e Sinalização de trânsito (9).
- Nesta mesma localização, há registros de natureza Mobilidade do tipo Acidente de trânsito (10)
- A localização $(-28.932, -49.470)$ registra uma maior parte de registros de Acidente de trânsito. A localização $(-28.932, -49.50)$ também apresenta ocorrências de Acidente de trânsito. Esta região pode ser considerada como crítica em trânsito.
- Ainda apresenta uma grande quantidade de ocorrência de Assalto nas proximidades de $(-28.933, -49.47)$ na metade do período analisado, e de $(-28.933, -49.50)$, durante o último quarto do período analisado.

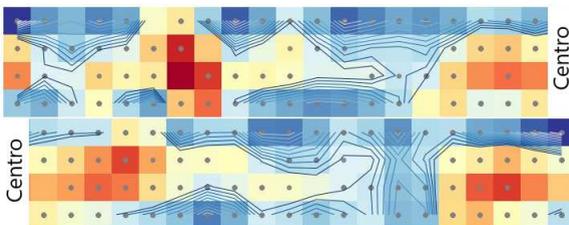
Figura 40 – Mapa de componentes do teste de aleatoriedade.



Fonte: Elaborado pelo autor.

Observando a *U-matrix* gerada pela biblioteca neste teste (Figura 41), percebe-se que ela foi expandida pela biblioteca `matplotlib` devido à grande variedade de dados. Comparando-a a Figura 42, percebe-se que os neurônios com coloração similar são organizados em *clusters*.

Figura 41 – *U-matrix* do teste de aleatoriedade.



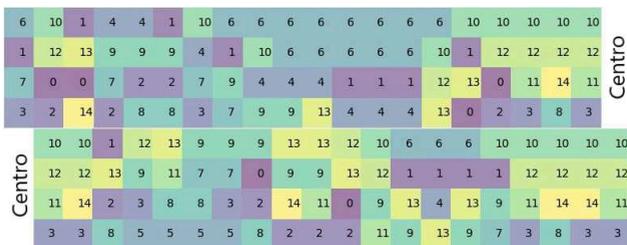
Fonte: Elaborado pelo autor.

Neste teste, devido à grande variância de localização e data, foram gerados mais clusters que no teste de ocorrência. Os centróides dos *clusters* também confirmam que a maior parte deles tem ocorrência entre 7 e 13.

Os centróides também podem servir como um modelo de previsão

a fim de antecipar a pertinência de novas ocorrências, assim como de interpretar o comportamento de uma região a partir da identificação de principais problemas registrados nela.

Figura 42 – *Clusters* do teste de aleatoriedade.



Fonte: Elaborado pelo autor.

Código 7 – Lista de clusters do teste de aleatoriedade

Topographic error = 0.699; Quantization error =
0.7959191510482715

Performing K-means SSE elbow sweep...

Centroid 0[-28.9537315 -49.47835081 1.66642134 9.18020649
0.44242564]

Centroid 1[-28.94592763 -49.49505284 2.4187612 12.00806101
0.34004193]

Centroid 2[-28.95129852 -49.48481155 1.23497136 6.98934274
0.36766273]

Centroid 3[-28.96224018 -49.49710276 1.15089152 5.82860983
0.52387472]

Centroid 4[-28.94378284 -49.48988496 2.09724542 11.23763539
0.20583519]

Centroid 5[-28.96772866 -49.51678875 1.01904832 3.64913566
0.29381591]

Centroid 6[-28.94478132 -49.49971836 2.88086463 13.23940271
0.24132442]

Centroid 7[-28.96071927 -49.49847281 1.50371066 8.34784402
0.22136422]

Centroid 8[-28.9562891 -49.49859785 1.08768174 5.11676312
0.37732224]

Centroid 9[-28.95301676 -49.50990427 1.82447257 10.10207424
0.30231148]

Centroid 10[-28.95405536 -49.49450002 2.68901428 12.67412069]

```

0.56811084]
Centroid 11[-28.9546095 -49.50058877 1.5577932 8.51842827
0.58922499]
Centroid 12[-28.95664831 -49.4955602 2.27852224 11.42309235
0.6118043 ]
Centroid 13[-28.9475147 -49.50496127 1.98807464 10.75892653
0.40473908]
Centroid 14[-28.95938044 -49.50101875 1.44046519 7.85741017
0.56329174]

```

6.3.5 Relatórios

Conforme explicado na Seção 6.3.2, os dados foram convertidos em um *Data Frame* com a ferramenta **pandas**. Esta ferramenta é uma biblioteca destinada a manipulação e análise de dados.

Ela também foi utilizada para gerar relatórios das ocorrências registradas. Neste trabalho, o relatório é o modo mais indicado para verificar as ocorrências mais registradas por local. Ou seja, o relatório descreve as ocorrências mais registradas numa localização precisa.

Diferentemente dos testes anteriores, este não utilizou latitudes e longitudes aleatórias, mas variou entre 25 localizações fixas pois, caso contrário, seria necessário gerar mais de um milhar de dados, devido a grande variedade - 14 tipos de ocorrências, 878 latitudes e 1205 longitudes, se considerar 4 casas decimais de precisão, o suficiente para esta aplicação. As latitudes e longitudes utilizadas são listadas na Tabela 7.

Tabela 7 – Latitudes e Longitudes.

Latitude	Longitude
-28.953752	-49.499440
-28.949340	-49.481930
-28.947669	-49.493968
-28.942636	-49.474163
-28.940515	-49.462822

Fonte: Elaborado pelo autor.

As principais ocorrências registradas neste teste são mostradas no Código 11.

 Código 8 – Relatório de ocorrências mais frequentes

location_lat	location_lon	ocorrencia	count
-28.9493	-49.4742	14	16
-28.9493	-49.4940	14	11
-28.9426	-49.4994	11	11
-28.9426	-49.4994	13	10
-28.9538	-49.4628	10	10
-28.9405	-49.4628	13	10
-28.9493	-49.4994	12	9
-28.9477	-49.4940	10	9
-28.9477	-49.4994	13	9
-28.9426	-49.4742	13	9
-28.9426	-49.4628	14	9
-28.9426	-49.4628	13	9
-28.9405	-49.4940	13	8
-28.9538	-49.4819	10	8
-28.9538	-49.4742	11	8
-28.9493	-49.4819	12	8
-28.9493	-49.4742	11	8
-28.9493	-49.4628	10	8
-28.9493	-49.4628	13	8
-28.9477	-49.4819	13	8
-28.9426	-49.4940	11	8
-28.9477	-49.4742	10	8
-28.9538	-49.4994	13	8
-28.9426	-49.4940	14	8
-28.9477	-49.4940	12	8
-28.9493	-49.4819	13	7
-28.9493	-49.4819	11	7
-28.9493	-49.4628	14	7
-28.9405	-49.4994	14	7
-28.9493	-49.4940	13	7

6.4 AVALIAÇÃO DA PLATAFORMA

Identificado como o segundo tipo de teste descrito na Seção 6.1, este teste avalia a comunicação da plataforma InterSCity efetuando cadastro de recursos e dados com o algoritmo de geração de dados, descrito na Seção 6.3.1.

Este foi um simples teste realizado a fim de demonstrar a taxa de sucesso de recebimento dos dados pela plataforma. Em cada etapa,

foi designado um tempo de espera após cadastrar um recurso e o envio de um dado. Cada recurso enviou uma quantidade estipulada de dados. Os resultados são mostrados na Tabela 8.

Tabela 8 – Taxa de sucesso de envio.

Recur- sos	Dados	Tempo após recurso (s)	Tempo após Dado (s)	Dados registra- dos
1	50	5	1	50
1	100	5	1	300
3	100	5	1	300
2	150	5	1	300
2	150	5	1	300
2	50	5	1	100
1	300	5	1	300
3	300	10	0	900
5	10	0	0	50
1	1000	0	0	1000

Fonte: Elaborado pelo autor.

A partir dos resultados exibidos na tabela, pode-se notar que a taxa de sucesso em todos os casos foi de 100% e independe da quantidade de recursos e dados e do tempo entre uma requisição e outra. Pode-se inferir, ainda, que todos os recursos foram registrados, pois foi possível registrar todos os dados. Caso tivesse havido falha no cadastro de algum recurso, os respectivos dados não teriam sido registrados, pois ele não teria um `uuid`.

6.5 DIFICULDADES

Ao longo do desenvolvimento do trabalho, foram encontradas dificuldades em cada etapa.

6.5.1 *Middleware*

A configuração de um *middleware* era uma etapa importante que definiria como o aplicativo iria funcionar. O FIWARE foi a primeira

opção de *middleware* considerada por ter funcionalidades desenvolvidas para diversas aplicações de *Smart City* e por ser amplamente utilizado em vários países. Entretanto, possui uma documentação muito grande que, por este mesmo motivo, nem sempre é clara e intuitiva. Portanto, cerca de 3 semanas foram dispendidas tentando instalá-lo, mas sem sucesso. Dependendo do tipo de aplicação de *Smart City*, seria necessário registrar-se no FIWARE Lab que, obrigatoriamente, necessita de uma declaração de grupo de trabalho, que não era o caso deste trabalho. Portanto, foi decidido tentar outro *middleware*.

A segunda opção foi a plataforma InterSCity. O primeiro problema ocorreu devido à incompatibilidade de versões do Docker e do Kong, uma *API Gateway* que direciona às portas de cada serviço. Muito tempo foi dedicado tentando resolver este problema, porém, por indicação da equipe técnica do InterSCity, a solução mais simples era trabalhar com a porta de cada serviço individualmente. O segundo problema aconteceu quando os containers do Docker corromperam por razão desconhecida e foi necessário recriar todas as imagens dos micros-serviços no servidor. Por isso, também foi preciso reenviar ocorrências para poder realizar os testes. Este problema não tem acontecido novamente desde então.

6.5.2 Desenvolvimento Android

Durante a etapa de desenvolvimento do aplicativo houve problemas na conciliação entre a versão mínima da API e algumas classes de solicitação HTTP e do serviço do GPS, que haviam sido descontinuadas. Entretanto, foi possível encontrar outras classes que solucionaram este problema.

6.5.3 Processamento de Dados

Na etapa de processamento de dados, percebe-se que a margem de erro se dá devido a média calculada pela biblioteca utilizando *k-means*. Portanto, os resultados dos centroides podem ser considerados somente em análise de regiões.

7 CONSIDERAÇÕES FINAIS

O desenvolvimento de uma arquitetura computacional para *Smart City* é um campo de estudo bastante importante, visto que as Tecnologias da Informação e Comunicação vem sendo utilizadas para esta finalidade em diversos tipos de aplicações. Buscar utilizá-las com a finalidade de expressar as principais solicitações dos cidadãos é um modo de interligar suas demandas e os líderes políticos da cidade onde vivem.

Por isso, explorando a popularização dos dispositivos móveis, foi desenvolvido o protótipo da aplicação. Deste modo, os cidadãos tem acesso à uma ferramenta onde é possível registrar as ocorrências da cidade em que vivem. Estas, em conjunto, indicam os principais problemas, estabelecem uma ligação entre suas demandas e os gestores e facilitam o processo de alocação de recursos da cidade.

A comunicação entre o aplicativo InterageAru e a plataforma InterSCity funcionou como esperado. O tempo de resposta não é considerado alto, comparado às simulações feitas com a API Postman, e não há perda de dados.

Apesar de os dados processados neste trabalho não terem sido enviados por cidadãos reais, foi possível concluir que informações inteligentes, características regionais e principais demandas podem ser identificadas com o auxílio de técnicas de aprendizagem não supervisionada e bibliotecas de análise de dados. Portanto, o emprego de uma aplicação como o InterageAru é viável. Entretanto, a implementação de um microserviço na plataforma InterSCity que retorne para o aplicativo os resultados obtidos pelos algoritmos apresentados é o ideal, a fim de tornar as informações transparentes.

A partir dos dados, pode-se também concluir que, devido aos atributos latitude e longitude, pode não haver uma relação linear entre o conjunto de dados, o que pode levar a um erro maior no cálculo dos centroides por parte da biblioteca utilizada. Ou seja, a biblioteca *sompy* é uma boa opção para demonstrar uma visão de características de pequenas regiões e não de localizações mais precisas, como as pertencentes à uma rua. Por isso, fez-se uso da biblioteca *pandas* a fim de gerar relatórios por localização e ocorrências registradas exatamente naquele local. Assim como os mapas de componentes e os centroides, gerados pelo SOM e pelo *k-means*, são interessantes para a análise por região, o relatório é vantajoso para identificar com precisão os locais que possuem uma maior demanda.

7.1 TRABALHOS FUTUROS

Há uma grande diversidade de tipos de aplicações em Sensoriamento Participativo. O desenvolvimento deste trabalho focou em explorar os eventos que acontecem em uma cidade. Entretanto, é possível expandir esta aplicação, bem como criar outras com diferentes propósitos, mas que explore a arquitetura proposta.

Algumas propostas de trabalhos futuros utilizando a arquitetura proposta são listadas a seguir:

1. Implementar outras funcionalidades
 - (a) Adicionar a visualização dos dados processados em um mapa no aplicativo.
 - (b) Utilizar algoritmos de aprendizado de máquina que oferecem uma diferente interpretação dos dados para processá-los.
 - (c) Adicionar outras opções de ocorrências conforme necessidade.
2. Amplificar a abrangência do aplicativo
 - (a) Implementar funcionalidades que enviem dados como localização e velocidade para a plataforma InterSCity a fim de estudar as principais trajetórias traçadas pelos cidadãos.
 - (b) Incrementar recursos de outras áreas de aplicação de *Smart City*.
3. Criar outras aplicações para plataformas Web e iOS, por exemplo.
4. Implementar um microsserviço de processamento de dados na plataforma, integrando os algoritmos apresentados, que responda às solicitações dos usuários.

REFERÊNCIAS

- ALBINO, V.; BERARDI, U.; DANGELICO, R. M. Smart Cities: Definitions, Dimensions, Performance, and Initiatives. *Journal of Urban Technology*, v. 22, n. 1, p. 3–21, 2015. ISSN 1063-0732. <<https://goo.gl/uCwfzZ>>.
- ALSWAILIM, M. A.; ZULKERNINE, M.; HASSANEIN, H. S. Classification of participatory sensing privacy schemes. *39th Annual IEEE Conference on Local Computer Networks Workshops*, p. 761–767, 2014. <<https://goo.gl/Ku9PJn>>.
- ALTOMARE, A. et al. Using clouds for smart city applications. *Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom*, v. 2, p. 234–237, 2013. ISSN 23302186.
- BALLAS, D. What makes a 'happy city'? *Cities*, v. 32, 2013. ISSN 02642751.
- BAPTISTA, M. e-Government and State Reform: Policy Dilemmas for Europe. *Electronic Journal of e-Government*, v. 3, n. 4, p. 167–174, 2005. <<http://www.ejeg.com/volume-3/vol3-iss4/v3-i4-art2.htm>>.
- BATTY, M. et al. Smart cities of the future. *European Physical Journal: Special Topics*, v. 214, n. 1, p. 481–518, 2012. ISSN 19516355.
- BAZZO, J. J.; MELGAREJO, D. C.; YOSHIMURA, R. S. Estratégias de Sensoriamento de Espectro Cooperativo Baseado em Autovalores. *III Workshop de Redes de Acesso em Banda Larga*, p. 102–115, 2013.
- BOGDAN-MARTIN, D. et al. *the State of Broadband 2017: Broadband Catalyzing Sustainable Development*. [s.n.], 2017. ISBN 9789261254018. <https://www.itu.int/dms_pub/itu-s/opb/pol/S-POL-BROADBAND.18-2017-PDF-E.pdf>.
- BURKE, J. A. et al. Participatory Sensing. 2006. <<http://escholarship.org/uc/item/19h777qd>>.
- BăTăGAN, L. Smart Cities and Sustainability Models. *Revista de Informatică Economică*, v. 15, n. 3, p. 80–87, 2011. ISSN 14531305. <<http://revistaie.ase.ro/content/59/07 - Batagan.pdf>>.

CARAGLIU, A.; Del Bo, C.; NIJKAMP, P. Smart Cities in Europe. *Journal of Urban Technology*, v. 18, n. 2, p. 65–82, 2011. ISSN 1063-0732. <<http://www.tandfonline.com/doi/abs/10.1080/10630732.2011.601117>>.

CELES, C. S. F. D. S. et al. Redes de Sensoriamento Participativo: Desafios e Oportunidades. In: *XXXIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Vitória, ES: SBRC, 2015. cap. 6.

CHEN, Y. et al. A vision of IoT: Applications, challenges, and opportunities with China Perspective. *IEEE Internet of Things Journal*, v. 2, n. 5, p. 1–1, 2014. ISSN 2327-4662. <<http://ieeexplore.ieee.org/document/6851114/>>.

CHOURABI, H. et al. Understanding smart cities: An integrative framework. *Proceedings of the Annual Hawaii International Conference on System Sciences*, n. January, p. 2289–2297, 2011. ISSN 15301605.

COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. *Sistemas Distribuídos - 4ed: Conceitos e Projeto*. BO-OKMAN COMPANHIA ED, 2007. ISBN 9788560031498. <<https://books.google.com.br/books?id=KSZ1rIRWmUoC>>.

CRANSHAW, J.; HONG, J. I.; SADEH, N. The Livelihoods Project : Utilizing Social Media to Understand the Dynamics of a City. *Icwsn*, p. 58–65, 2012. ISSN 0021-9193.

CRISTOFARO, E. D.; SORIENTE, C. Participatory privacy: Enabling privacy in participatory sensing. *IEEE Network*, v. 27, n. 1, p. 32–36, 2013. ISSN 08908044.

ECKHOFF, D.; WAGNER, I. Privacy in the Smart City - Application, Technologies, Challenges and Solutions. *IEEE Communications Surveys & Tutorials*, n. c, p. 1–28, 2017. ISSN 1553-877X.

EISENMAN, S. B. et al. The BikeNet mobile sensing system for cyclist experience mapping. *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys '07)*, p. 87, 2007. ISSN 1550-4859. <<http://portal.acm.org/citation.cfm?doid=1322263.1322273>>.

ESTRIN, D. Participatory sensing: applications and architecture [Internet Predictions]. *IEEE Internet Computing*, v. 14, n. 1, p. 12–42, 2010. ISSN 1089-7801. <<https://goo.gl/WPcJYu>>.

FAROKHI, F.; SHAMES, I. Preserving Privacy of Agents in Participatory-Sensing Schemes for Traffic Estimation. n. Cdc, p. 6739–6744, 2016.

FREITAS, M. P. de. *Sensoriamento Participativo: conceito, desafios e oportunidades*. Tese (Doutorado) — Faculdade Batista do Estado do Rio de Janeiro, 2015. <<https://goo.gl/gDo9RB>>.

GHARAIBEH, A. et al. Smart Cities: A Survey on Data Management, Security and Enabling Technologies. *IEEE Communications Surveys & Tutorials*, X, n. X, p. 1–1, 2017. ISSN 1553-877X. <<http://ieeexplore.ieee.org/document/8003273/>>.

GIFFINGER, R. Smart cities Ranking of European medium-sized cities. *October*, v. 16, n. October, p. 13–18, 2007. ISSN 02642751. <<http://linkinghub.elsevier.com/retrieve/pii/S026427519800050X>>.

GROAT, M. M. et al. Enhancing privacy in participatory sensing applications with multidimensional data. *2012 IEEE International Conference on Pervasive Computing and Communications, PerCom 2012*, n. March, p. 144–152, 2012.

GUO, B. et al. From participatory sensing to Mobile Crowd Sensing. *2014 IEEE International Conference on Pervasive Computing and Communication Workshops, PERCOM WORKSHOPS 2014*, p. 593–598, 2014.

HOLLANDS, R. G. Will the real smart city please stand up? *City*, v. 12, n. 3, p. 303–320, 2008. ISSN 1360-4813. <<https://www.tandfonline.com/doi/full/10.1080/13604810802479126>>.

HUSSIN, M. F. B. et al. Accident reporting system using an iOS application. *Proceedings - 2014 IEEE Conference on System, Process and Control, ICSPC 2014*, n. December, p. 13–18, 2014.

IBGE. *Acesso à Internet e à televisão e posse de telefone móvel celular para uso pessoal*. Dezembro 2016. Pesquisa Nacional por Amostra de Domicílios (Pnad). <<https://biblioteca.ibge.gov.br/visualizacao/livros/liv99054.pdf>>. Acessado em 16 nov. 2017.

INDRA. *Conceitos SOFIA2*. [S.l.], 2014. <[http://sofia2.com/docs/\(PT\)SOFIA2-ConceitosSOFIA2.pdf](http://sofia2.com/docs/(PT)SOFIA2-ConceitosSOFIA2.pdf)>. Acessado em 10 nov. 2017.

KANHERE, S. S. Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, v. 7753 LNCS, p. 19–26, 2013. ISSN 03029743.

KHOI, N. M.; RODR, L.; CASTELEYN, S. CitizenSense - A generic user-oriented participatory sensing framework. 2017.

KOHONEN, T. *Self-Organizing Maps*.

Springer, 2000. ISBN 3540679219.

<<https://www.amazon.com/Self-Organizing-Maps-Teuvo-Kohonen/dp/3540679219?SubscriptionId=0JYN1NVW651KCA56C102tag=techkie-20linkCode=xm2camp=2025creative=165953creativeASIN=3540679219>>.

KOMNINOS, N.; PALLOT, M.; SCHAFFERS, H. Special Issue on Smart Cities and the Future Internet in Europe. *Journal of the Knowledge Economy*, v. 4, n. 2, p. 119–134, 2013. ISSN 18687873.

LONG, X.; JIN, L.; JOSHI, J. Exploring trajectory-driven local geographic topics in foursquare. *Proceedings of the 2012 ACM Conference on Ubiquitous Computing - UbiComp '12*, p. 927, 2012. <<http://dl.acm.org/citation.cfm?doid=2370216.2370423>>.

M. Del Esposte, A. et al. InterSCity: A Scalable Microservice-based Open Source Platform for Smart Cities. *Proceedings of the 6th International Conference on Smart Cities and Green ICT Systems*, p. 35–46, 2017. <<https://goo.gl/7vwXwr>>.

MAISONNEUVE, N. et al. NoiseTube: Measuring and mapping noise pollution with mobile phones. *Environmental Science and Engineering (Subseries: Environmental Science)*, p. 215–228, 2009. ISSN 18635520.

MEIJER, A. J.; GIL-GARCIA, J. R.; BOLÍVAR, M.

P. R. Smart City Research. *Social Science Computer Review*, v. 34, n. 6, p. 647–656, 2016. ISSN 0894-4393.

<<http://journals.sagepub.com/doi/10.1177/0894439315618890>>.

MEIRELLES, F. S. *28ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas*. Abril 2017. Fundação Getúlio Vargas de São Paulo. <<http://www.fgv.br/cia/pesquisa>>. Acessado em 16 nov. 2017.

MITTON, N. et al. Combining Cloud and sensors in a smart city environment. *EURASIP Journal on Wireless Communications and Networking*, v. 2012, n. 1, p. 247, 2012. ISSN 1687-1499. <<https://jwcn-urasipjournals.springeropen.com/articles/10.1186/1687-1499-2012-247>>.

MOHAN, P.; PADMANABHAN, V. N.; RAMJEE, R. Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones. *Proceedings of the 6th ACM conference on Embedded network sensor systems - SenSys '08*, p. 323, 2008. <<https://goo.gl/X9gGuL>>.

MUN, M. U. et al. PEIR, the personal environmental impact report, as a platform for participatory sensing systems research. *MobiSys*, p. 55–68, 2009. <<http://dl.acm.org/citation.cfm?id=1555823>>.

NAM, T.; PARDO, T. A. Conceptualizing smart city with dimensions of technology, people, and institutions. *Proceedings of the 12th Annual International Digital Government Research Conference on Digital Government Innovation in Challenging Times - dg.o '11*, p. 282, 2011. ISSN 1450307620. <<http://dl.acm.org/citation.cfm?doid=2037556.2037602>>.

PAN, G. et al. Trace analysis and mining for smart cities: Issues, methods, and applications. *IEEE Communications Magazine*, v. 51, n. 6, p. 120–126, 2013. ISSN 0163-6804. <<http://ieeexplore.ieee.org/document/6525604/>>.

RANA, R. K. et al. Ear-Phone : An End-to-End Participatory Urban Noise Mapping System. *Proceedings of the International Conference on Information Processing in Sensor Networks IPSN*, p. 105–116, 2010. <<http://portal.acm.org/citation.cfm?doid=1791212.1791226>>.

REDDY, S. et al. Image Browsing , Processing , and Clustering for Participatory Sensing : Lessons From a DietSense Prototype. *EmNets'07*, p. 13–17, 2007.

RUGE, L.; ALTAKROURI, B.; SCHRADER, A. SoundOfTheCity - Continuous noise monitoring for a healthy city. *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2013 IEEE International Conference on*, n. March, p. 670–675, 2013.

SEI, Y.; OHSUGA, A. Privacy Preservation for Participatory Sensing Applications. 2016.

SILVA, T. H. et al. Challenges and opportunities on the large scale study of city dynamics using participatory sensing. *2013 IEEE Symposium on Computers and Communications (ISCC)*, n. July 2014, p. 000528–000534, 2013. ISSN 15301346. <<http://ieeexplore.ieee.org/document/6755000/>>.

SMARTMETROPOLIS. *Requisitos e Plataformas de Middleware para Cidades Inteligentes*. Natal, 2016. <<http://smartmetropolis.imd.ufrn.br/wp-content/uploads/2016/05/RT1-WP5.pdf>>. Acessado em 9 nov. 2017.

SMARTMETROPOLIS. *Relatório de Atividades do Sexto Trimestre*. Natal, 2017. <<http://smartmetropolis.imd.ufrn.br/wp-content/uploads/2017/11/RT6-WP2-app.pdf>>. Acessado em 27 nov. 2017.

SU, K.; LI, J.; FU, H. Smart city and the applications. *2011 International Conference on Electronics, Communications and Control, ICECC 2011 - Proceedings*, p. 1028–1031, 2011. ISSN 2327-4662.

UNDESA. *World Urbanization Prospects*. [s.n.], 2014. 32 p. ISSN 1435-9871. ISBN 9789211515176. <<http://esa.un.org/unpd/wup/Highlights/WUP2014-Highlights.pdf>>.

United Nations. World Population Prospects: The 2012 Revision. Highlights and Advance Tables. *Population and development review*, v. 36, p. 775–801, 2013. ISSN 0098-7921. <<http://esa.un.org/unpd/wpp/Documentation/publications.htm>>.

VAIDYA, S. et al. People as Sensors for Smart City. *International Journal of Computer Applications*, v. 116, n. 18, p. 28–32, 2015.

VERGARA-LAURENS, I. J.; MENDEZ, D.; LABRADOR, M. A. Privacy, quality of information, and energy consumption in Participatory Sensing systems. *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, p. 199–207, 2014. <<http://ieeexplore.ieee.org/document/6813961/>>.

VILLANUEVA, J. et al. Civitas : The Smart City Middleware , from Sensors to Big Data. n. 3, 2013.

XIAO, Z.; LIM, H. B.; PONNAMBALAM, L. Participatory Sensing for Smart Cities: A Case Study on Transport Trip Quality Measurement. *IEEE Transactions on Industrial Informatics*, v. 13, n. 2, p. 759–770, 2017.

ZANELLA, a. et al. Internet of Things for Smart Cities. *IEEE Internet of Things Journal*, v. 1, n. 1, p. 22–32, 2014. ISSN 2327-4662. <<https://goo.gl/vLqGdn>>.

APÊNDICE A - Algoritmos utilizados no processamento de dados

Código 9 – Salva os dados em formato .csv

```
import requests, json, csv, unicodecode
import numpy as np

#Converts the string to an index
def index_natureza(s_nat):
    if (s_nat == 'Infraestrutura'):
        i_nat = 1
    elif (s_nat == 'Mobilidade'):
        i_nat = 2
    else:
        i_nat = 3
    return i_nat

#Converts the string to an index
def index_ocorrencia(s_ocorr):
    ocorrencias_infra = ['Buraco no pavimento', 'Calçada sem
        manutencao', 'Desmoronamento', 'Poste caído',
        'Iluminacao deficiente', 'Vazamento de agua',
        'Enchente', 'Queda de energia', 'Sinalizacao de
        transito']
    ocorrencias_mob = ['Acidente de transito', 'Obra',
        'Congestionamento']
    ocorrencias_seg = ['Assalto', 'Furto']

    if (s_ocorr) in ocorrencias_infra:
        i_ocorr = ocorrencias_infra.index(s_ocorr) + 1
    elif (s_ocorr) in ocorrencias_mob:
        i_ocorr = ocorrencias_mob.index(s_ocorr) + 10
    else:
        i_ocorr = ocorrencias_seg.index(s_ocorr) + 13
    return i_ocorr

#Request POST to the url
url = "http://150.162.234.28:4000/resources/data"

dat = {'capabilities': ['participatory_sensing']}
headers = {'content-type': 'application/json'}
req_res = requests.post(url, data=json.dumps(dat),
    headers=headers)
parsed_json = json.loads(req_res.text)
data_json = parsed_json['resources']
```

```

file = open('data.csv', 'w')
csvwriter = csv.writer(file)

#Create csv headers
csvwriter.writerow(['location_lat', 'location_lon', 'natureza',
                    'ocorrencia', 'date'])

#Write every row to the csv
for resource in data_json:
    data_json_resource =
        resource['capabilities']['participatory_sensing']
    for dat in data_json_resource:
        nat =
            index_natureza(unidecode.unidecode(dat['natureza']))
        occur =
            index_ocorrencia(unidecode.unidecode(dat['ocorrencia']))
        csvwriter.writerow([dat['location']['lat'],
                            dat['location']['lon'], nat, occur, dat['date']])

file.close()

#Shows the csv
with open('data.csv', 'r') as csvfile:
    spamreader = csv.reader(csvfile, delimiter=',', quotechar='|')
    for row in spamreader:
        print ', '.join(row)

```

Código 10 – Processa dados sobre .csv

```

import requests, json, csv, unidecode
import numpy as np
from sompy.sompy import SOMFactory
from sompy.visualization.mapview import View2D
from sompy.visualization.umatrix import UMatrixView
from sompy.visualization.hitmap import HitMapView

import pandas
from sklearn.preprocessing import MinMaxScaler

# converte datetime em string do csv lido para datetime
series = pandas.read_csv('data.csv', usecols=['date'])
series = pandas.to_datetime(series['date'], format='%Y-%m-%d
%H:%M:%S')

```

```

# prepare data for normalization
values = series.values
values = values.reshape((len(values), 1))
# train the normalization
scaler = MinMaxScaler(feature_range=(0, 1))
scaler = scaler.fit(values)
print(scaler)
# normalize the dataset and print the rows
normalized = scaler.transform(values)

#Delete the date column
df = pandas.read_csv('data.csv')
df = df.drop('date', axis=1)
df = df.assign(date = normalized)

print(df)

names = ['Latitude', "Longitude", 'Natureza','Ocorrencia',
         'Data']
sm = SOMFactory().build(df.values, normalization = 'var',
                        initialization='pca', component_names=names)
sm.train(n_job=1, verbose=False, train_rough_len=2,
        train_finetune_len=5)

topographic_error = sm.calculate_topographic_error()
quantization_error = np.mean(sm._bmu[1])
print ("Topographic error = %s; Quantization error = %s" %
      (topographic_error, quantization_error))

# Component map
view2D = View2D(10,10,"rand data",text_size=12)
view2D.show(sm, col_sz=4, which_dim="all", desnormalize=True)

# U-matrix
umat = UMatrixView(width=20,height=20,title='U-matrix')
umat.show(sm)

# Clustering
K = 20 #
K_opt = 5
[labels, km, norm_data] = sm.cluster(K, K_opt)
hits = HitMapView(20,20,"Clustering",text_size=12)
a=hits.show(sm)

```

 Código 11 – Relatório sobre dados do .csv

```

import requests, json, csv, unicode
import numpy as np
from sompy.sompy import SOMFactory
import folium

import pandas
from sklearn.preprocessing import MinMaxScaler

df = pandas.read_csv('data.csv')
#Delete the date column
df = df.drop('date', axis=1)
#Delete the natureza column
df = df.drop('natureza', axis=1)
df = df.round({'location_lat': 4, 'location_lon': 4})

df_most_freq = df.groupby(['location_lat', 'location_lon',
                           'ocorrencia']).size().reset_index(name='count')
print(df_most_freq.sort_values(['count'], ascending=False))

```

 Código 12 – Gerador de recursos e dados aleatórios

```

import requests, json, datetime
from random import randint, uniform
from random import randrange
from datetime import timedelta
import time

#constants used to create several resources and their data
N_RESOURCES = 1
N_DATA_PER_RESOURCES = 5

#function that creates a random CPF
#it returns a string
def createCPF():

    cpf = []
    sum = 0

    for i in range(9):
        cpf.append(randint(0,9))
        sum = sum + (cpf[i] * (i+2))

```

```

resto_1 = sum%11

if (resto_1 < 2):
    digit_1 = 0
else:
    digit_1 = 11 - resto_1

sum = 0
sum = digit_1 * 2

for i in range(9):
    sum = sum + (cpf[i] * (i+3))

resto_2 = sum%11

if (resto_2 < 2):
    digit_2 = 0
else:
    digit_2 = 11 - resto_2

cpf.reverse()
cpf_str = ''.join(map(str, cpf))
cpf_str = cpf_str + str(digit_1) + str(digit_2)

return cpf_str

#function that creates the JSON of a resource
def createResource():

    cpf_s = createCPF()

    resource_object = {
        "data": {
            "description": cpf_s,
            "capabilities": [
                "participatory_sensing"
            ],
            "status": "active",
            "lat": uniform(-28.912090, -28.999855),
            "lon": uniform(-49.436552, -49.556962)
        }
    }

```

```

return resource_object

def random_date(date_start, date_end):

    interval = date_end - date_start
    int = (interval.days * 24 * 60 * 60) + interval.seconds
    random_second = randrange(int)
    return date_start + timedelta(seconds=random_second)

d1 = datetime.datetime.strptime('1/1/2018 12:00 PM', '%m/%d/%Y
    %I:%M %p')
d2 = datetime.datetime.strptime('4/30/2018 12:00 PM', '%m/%d/%Y
    %I:%M %p')

#function that creates the JSON of data to be sent
def createData():

    naturezas = ['Infraestrutura', 'Mobilidade', 'Seguranca']
    ocorrencias_infra = ['Buraco no pavimento', 'Calçada sem
        manutencao', 'Desmoroamento', 'Poste caído', 'Iluminacao
        deficiente', 'Vazamento de agua', 'Enchente', 'Queda de
        energia', 'Sinalizacao de transito']
    ocorrencias_mob = ['Acidente de transito', 'Obra',
        'Congestionamento']
    ocorrencias_seg = ['Assalto', 'Furto']

    index_nat = randint(0,len(naturezas)-1)

    if (index_nat == 1):
        ocorrencia =
            ocorrencias_mob[randint(0,len(ocorrencias_mob)-1)]
    elif(index_nat == 2):
        ocorrencia =
            ocorrencias_seg[randint(0,len(ocorrencias_seg)-1)]
    else:
        ocorrencia =
            ocorrencias_infra[randint(0,len(ocorrencias_infra)-1)]

    data_object = {
        "data": [
            {
                "location": {
                    "lat": uniform(-28.912090, -28.999855),

```

```

        "lon": uniform(-49.436552, -49.556962)
    },
    "natureza": naturezas[index_nat],
    "ocorrencia": ocorrencia,
    "timestamp": ((random_date(d1,
        d2)).strftime('%Y-%m-%dT%H:%M:%S.000Z'))
    }
    ]
}
return data_object

soma = 0

for i in range (N_RESOURCES):
    resource_obj = createResource()
    url_resource = "http://150.162.234.28:3002/resources"
    headers = {'content-type': 'application/json'}
    req_res = requests.post(url_resource,
        data=json.dumps(resource_obj), headers=headers)
    parsed_json = json.loads(req_res.text)
    uuid = parsed_json['data']['uuid']
    time.sleep(0)

for j in range(N_DATA_PER_RESOURCES):
    data_obj = createData()
    time.sleep(0)
    url_data = "http://150.162.234.28:3002/resources/" + uuid +
        "/data/participatory_sensing"
    req_data = requests.post(url_data,
        data=json.dumps(data_obj), headers=headers)
    if (req_data.status_code == 201):
        print('OK')
        soma = soma + 1

print(soma)

```

ANEXO A - Documentação do InterSCity

As partes principais da documentação da plataforma InterSCity utilizada neste trabalho são inseridas neste Anexo. Para uma leitura mais completa, acesse <https://gitlab.com/smart-city-software-platform/docs.git>.

A.1 OVERVIEW

The project aims at providing high-level services to support novel applications that interact with city's resources such as bus, street cameras, environmental sensors, and public open data. For this purpose, it works as an integrated, scalable software infrastructure gathering three key-enabling technologies: IoT, BigData and Cloud Computing. Such infrastructure must support cross-domain applications rather than technological silos by sharing services, APIs, data, and standards.

The Platform provides high-level RESTful services to support the development of smart city applications, services and tools for different purposes and domains. Currently, it is composed of six different microservices that provide features for the integration of IoT devices (Resource Adaptor), data and resource management (Resource Catalog , Data Collector and Actuator Controller), resource discovery through context data (Resource Discovery) and visualization (Resource Viewer)

The platform intermediates all communication between smart city applications and IoT devices, as well as the complexity of city-scale data management. Specific IoT Gateways register city resources and post their sensorized data to the platform through the Resource Adaptor API. IoT Gateways may also subscribe to receive notifications to act over undelaying actuator devices

Next section describes in more detail the platform's microservices architecture, designed to provide the required features to support Smart Cities.

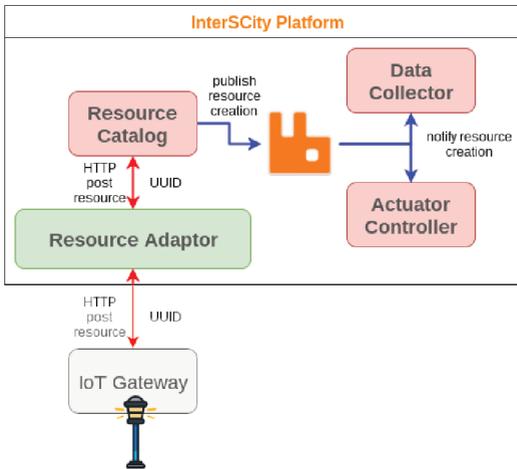
A.2 MICROSERVICES VIEW

In order to properly provide an unified technological infrastructure to city-scale services, the platform needs to integrate a large number of heterogeneous physical devices and services. Thus, the platform is based on a scalable, distributed microservices architecture. To see

the detailed relation of existing microservices, check the Microservices documentation page

Microservices can communicate through their RESTful APIs or asynchronously through RabbitMQ message bus with the publish-subscribe design pattern. In particular, we vastly use the topic-based communication through RabbitMQ as explained in this tutorial. To demonstrate how the microservices communicate with each other, we present below the main scenarios of platform usage that trigger the execution of more than one platform service. In all diagrams, blue arrows represent asynchronous communication through AMQP, while red arrows represent HTTP-based communication through REST APIs.

Figure 43 Microservices communication on resource creation.



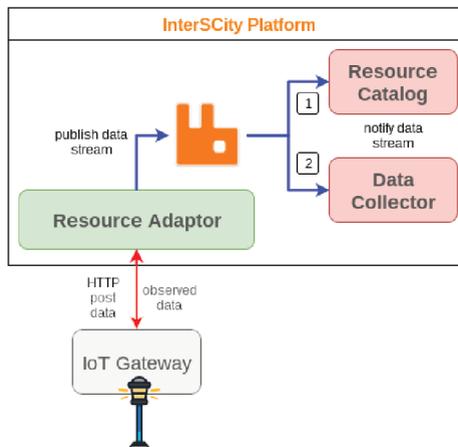
1. Physical devices integration: City's resources are coupled with cyber-physical devices, such as sensors and actuators, which may integrate to the platform by several different protocols. Thus, the Resource Adaptor is a unified endpoint service to which devices can register at the platform, send and request data. Every registered resource receives a UUID. Check out the supported protocols and technical details in the Resource Adaptor page.

2. Resource register: In order to make a resource available on the platform, Resource Adaptor sends the resource meta-data

to Resource Cataloguer service through its REST API. These meta-data describes the main features of a resource, exposing its capabilities, location and other important information.

3.Resource creation notification: After registering a new resource, the Resource Cataloguer publish an event to the RabbitMQ message bus which may notify the Data Collector service if the resource has sensor capabilities. Similarly, the Actuator Controller service is notified whenever a new resource has actuator capabilities.

Figura 44 Microservices communication on data stream.



1.Collecting data from resources: An IoT Gateway post new sensor data to the platform through the Resource Adaptor REST API. Then, the Resource Adaptor publishes a new event to RabbitMQ in the data_stream topic.

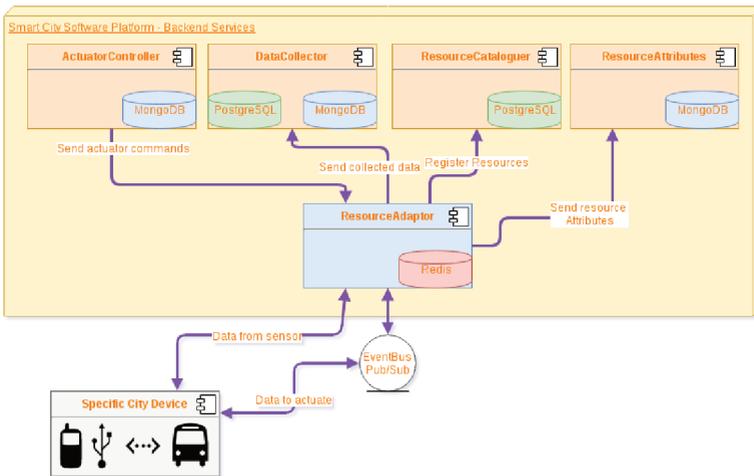
2.Data stream notification: Microservices interested in the most recent data provided by resources must subscribe to receive notifications. Currently, the Data Collector service stores all observed data in a historical database for futher processing. In a near future, we intended to:

- Implement the location update on Resource Catalog
- Implement a Big Data architecture for real time processing

A.2.1 Resource Adaptor

Resource Adaptor is the codebase Rails API application to create resources to encapsulate or simulate sensors/actuators for the Smart City Platform. In this page we document what is a Resource Adaptor and how you can create your own resources to be used by your app in the platform.

Figura 45 Design roadmap.



What is the Resource Adaptor?

The Resource Adaptor works as an Adapter of the city's sensors/actuators for the platform services. A Resource Adaptor communicates directly with the physical device and encapsulates specificities of communication protocols and data formatting related to the device. The main objectives of the Resource Adaptor are:

- Encapsulate/Simulate the communication with devices
- Manage all the interaction with the device: how to collect data, how to store information, when to send request to actuators
- Manage the registering of components in the Smart City Platform
- Expose a service API to allow the Resource Directory to ask for metadata and status of components

- Expose a standardized API to allow the Data Collector ask to the latest data of components or send requests to actuators

Thus, a Resource Adaptor is a service that can encapsulate one or more resource.

What is a Resource?

A resource is literally a “Thing” in the world, if you think in Internet of Things terms. In practice, each resource will be a sensor, an actuator or a micro-controller with many sensors and actuators. Two of the main objectives of the Smart City Platform are to allow applications to access and manipulate the resources of a city and to abstract implementation details of physical devices. In this sense, no matter the specific type of two temperature sensors, even if they capture temperatures in different ways, using different technologies, this data should be provided standardly for applications and users.

In order to become accessible in the Smart City Platform, a Resource Adaptor must register its resources in the Resource Cataloguer. Among other data, each resource must send the basic URL to access its REST API. Let’s consider the case when we have a Resource Adaptor encapsulating a network of USP’s sensors. Each sensor is a resource that must be registered in the platform to become accessible.

A.2.2 Data Collector

Data Collector’s main goal is to provide methods to search data collected from Resources.

This service is used both by applications and the Catalog Service, and offers developers access to information coming from different sensors scattered throughout the city. For example, with this service one could obtain data of all temperature sensors on USP.

Requests to this service usually involve receiving or sending a data structure in JSON format.

All interactions with the Data Collector are done through REST requests. Field description:

- uuid:** A non-quoted String corresponding to the resource’s uuid (not to be confused with the local id assigned by the Data Collector!). May have been applied beforehand by the Discovery, or stored directly by the client’s application.

- capabilities:** Capabilities are capacity that resources are able to respond to which we keep the data.
- value:** The collected value. It can be an number (float,integer) or string.
- date:** This field respresent the date when the data was collected.

A.2.3 Resource Catalog

Component of the middleware for smart cities that aggregates all resources' (sensors and actuators) metadata required by all platform services. It provides data, such as status, configuration, location, and identification for each registered sensor and actuator in the smart cities platform. All resources in the platform must be registered in this system.

Figura 46 Architecture.

