

UNIVERSIDADE DO VALE DO RIO DOS SINOS - UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL MESTRADO

FÁBIO VIEGAS

**UFOLLOWER: UM MODELO PARA CIDADES INTELIGENTES
BASEADO EM SEGURANÇA E VIGILÂNCIA UBÍQUAS**

São Leopoldo

2020

Fábio Viegas

**UFOLLOWER: UM MODELO PARA CIDADES INTELIGENTES
BASEADO EM SEGURANÇA E VIGILÂNCIA UBÍQUAS**

Dissertação apresentada como requisito parcial
para a obtenção do título de Mestre, pelo
Programa de Pós-Graduação em Computação
Aplicada da Universidade do Vale do Rio dos
Sinos – UNISINOS

Orientador:

Prof. Dr. Jorge Luis Victória Barbosa

Co-orientador:

Prof. Dr. Rafael Kunst

São Leopoldo

2020

V656u Viegas, Fábio

UFollower : um modelo para cidades inteligentes baseado em segurança e vigilância ubíquas / por Fábio Viegas. – 2020.

95 f. : il., 30 cm.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2020.

Orientação: Prof. Dr. Jorge Luis Victória Barbosa ;

Coorientação: Prof. Dr. Rafael Kunst.

1. Cidades inteligentes – Modelo. 2. Teoria Bayesiana de Decisão Estatística. 3. Computação ubíqua. 4. Segurança pública. I. Título.

CDU 004.056

Catálogo na Fonte:
Bibliotecária Vanessa Borges Nunes - CRB 10/1556

RESUMO

O avanço tecnológico possibilita o desenvolvimento de soluções de tecnologia da informação e comunicação (ICT do inglês *Information and Communication Technology*) desde um nicho simples, aplicado a um contexto específico, como a informatização de ambientes macros, como uma cidade por exemplo. Há diversos objetivos que podem ser alcançados para esse último caso. Para citar alguns: assistência a recursos de segurança (câmeras e sensores) e qualificar ambientes de segurança (mapas de calor identificando zonas de risco). O presente trabalho apresenta o modelo UFollower (*Ubiquitous Follower*) o qual atende a esse cenário. O modelo UFollower propõe o uso da computação ubíqua como suporte à segurança. O mesmo contempla uma ontologia para pessoas, recursos e ambientes. Junto da ontologia é proposto um sistema multiagente para realizar as inferências e suporte à segurança. Utilizando mecanismos de predição e históricos de contextos, foi possível alcançar a taxa de 70,35% e acerto na inferência de ocorrência de um crime. Uma contribuição diferencial do modelo é o acompanhamento ubíquo do cidadão, assistindo-o de forma onipresente através de dados do contexto ao qual está inserido e utilizando informações as quais o modelo julgar úteis de acordo com o perfil cadastrado. A comparação com os trabalhos relacionados indica que o UFollower é a única proposta que apresenta a questão da Segurança Pública, otimizando a ação de agentes através de inferências realizadas através de Redes Bayesianas, geolocalização e histórico de contexto.

A avaliação se apresenta através de processos definidos na forma de experimentos descritos no formato de cenários, os quais têm como objetivo avaliar as hipóteses relacionadas a cada um deles. O ponto inicial para a formulação de cada uma das hipóteses é o fato de que se tem uma ideia de uma causa e o efeito relacionado a ela. Um protótipo foi implementado e aplicado em uma avaliação organizada em três cenários, onde foi possível comprovar a relevância da criação do modelo. Ao efetuar as avaliações, conclui-se que o modelo UFollower amplia as funcionalidades dos entes de segurança, passando a oferecer suporte e monitoramento ubíquos simultaneamente, fornecendo *feedbacks* e recomendações ao perfil que o esteja usando com o intuito de ajudá-lo a acompanhar seu deslocamento e segurança particular de forma contínua.

Palavras-Chave: Modelo. Cidades Inteligentes. Teoria Bayesiana de Decisão Estatística. Computação Ubíqua, Segurança Pública.

ABSTRACT

Technological advancement enables the development of information and communication technology solutions (ICT from English Information and Communication Technology) from a simple niche, applied to a specific context, such as the computerization of macro environments, a city for example. There are several objectives that can be achieved for the latter case. To name a few: assistance with security features (cameras and sensors) and qualifying security environments (heat maps identifying risk zones). The present work presents the UFollower model (Ubiquitous Follower) which meets this scenario. The UFollower model proposes the use of ubiquitous computing to support security. It includes an ontology for people, resources and environments. Along with the ontology, a multiagent system is proposed to make inferences and support security. Using prediction mechanisms and historical contexts, it was possible to reach the rate of 70.35% and correct the inference of the occurrence of a crime. A differential contribution of the model is ubiquitous monitoring of the citizen, watching him in an ubiquitous way through data from the context to which he is inserted, as well as using information that the model deems useful according to the registered profile. The comparison with related works indicates that UFollower is the only proposal that presents the issue of Public Security, optimizing the action of agents through inferences made through Bayesian Networks.

The evaluation is presented through processes defined in the form of experiments described in the scenario format, which aim to evaluate the hypotheses related to each one of them. The starting point for formulating each hypothesis is the fact that you have an idea of a cause and the effect related to it. A prototype was implemented and applied in an evaluation organized in three scenarios, where it was possible to prove the relevance of creating the model.

Keywords: *Model. Smart Cities. Bayesian Theory of Statistical Decision. Ubiquitous Computing, Public Security.*

LISTA DE FIGURAS

Figura 1: Conceitos do modelo UFollower	19
Figura 2: Elementos representativos da computação ubíqua.....	21
Figura 3: Elementos e características da computação ubíqua	22
Figura 4: Representação da arquitetura de computação em nuvem em camadas <i>Front End</i> e <i>Back End</i>	23
Figura 5: Pilares da Internet das coisas	24
Figura 6: Diagrama de blocos de computador vestível típico	26
Figura 7: Monitoramento por tornozeleira eletrônica no UFollower	27
Figura 8- <i>Wearable</i> , Nodo Pessoal e UFollowerNode monitorando batimentos cardíacos	27
Figura 9: Classificação das Cidades	31
Figura 10: Principais componentes de uma cidade inteligente.....	35
Figura 11: Camadas do modelo IDS.....	36
Figura 12: Arquitetura SPATH.....	37
Figura 13: Ilustração de resumos e transmissões de vídeo	37
Figura 14: Estrutura do aplicativo Campus USP	39
Figura 15: Arquitetura do sistema <i>Urban Safety</i>	40
Figura 16: Arquitetura <i>smart surveillance</i>	42
Figura 17: Visão geral do modelo UFollower	48
Figura 18: Arquitetura do UFollower	50
Figura 19 - Componente de Contexto como parte do Agente de Recomendação	51
Figura 20: Descrição do UFollowerServer	51
Figura 21: Ontologia de condições de segurança	53
Figura 22- Símbolos da notação Prometheus empregado <i>plugin</i> PDT	54
Figura 23: Sistema Multiagente do modelo UFollower	55
Figura 24: Agente para Perfil do modelo UFollower	55
Figura 25: Agente para Monitoramento do modelo UFollower	56
Figura 26: Agente para Recomendação do modelo UFollower	56
Figura 27: Agente para Buscas do modelo UFollower.....	57
Figura 28: Agente para Alertas do modelo UFollower	57
Figura 29: Arquitetura do Aplicativo Móvel e do <i>Site Web</i> do modelo UFollower.....	59
Figura 30: Pacotes básicos do protótipo implementado	61
Figura 31: Representação do contexto interagindo com o Servidor.....	62
Figura 32: Quantidade de Registros para inferência de Redes Bayesianas.....	63
Figura 33: Mapa do Femicídio	66

Figura 34: Observatório da Violência contra a Mulher da SSP-RS	67
Figura 35: Trecho da implementação dos modelos NB do Scikit Learn.....	69
Figura 36: Fluxo do uso do UFollower App.....	70
Figura 37: Aplicativo UFollower Nodo Pessoal.....	74
Figura 38: Perfil Morador no App.....	74
Figura 39: Módulo <i>Web</i> do Modelo UFollower	71
Figura 40: Representação de consumo da API REST no nodo pessoal	72
Figura 41: API completa para utilização do UFollower.....	72
Figura 42: API disponibilizada pela retaguarda <i>Web</i> do modelo UFollower(a)	73
Figura 43: API disponibilizada pela retaguarda <i>Web</i> do modelo UFollower(b)	73
Figura 44: Perfil Agente sendo notificado em seu Nodo Pessoal.....	74
Figura 45: Morador_2 caminhando sendo interceptado pelo Agente antes de se aproximar mais do Morador_1	75
Figura 46: Morador_2 já vinha sendo monitorado antes da emissão do alerta	78
Figura 47: Relação de crimes por bairros em Porto Alegre	80
Figura 48: Relação de crimes com os dias da semana.....	80
Figura 49: Relação crimes com os turnos do dia.....	81
Figura 50: Perfil turista no aplicativo	82
Figura 51: Turista_1 sendo notificada	82
Figura 52: Notificação para a retaguarda do modelo.	83
Figura 53: Identificação da área limítrofe da região perigosa	84

LISTA DE TABELAS

Tabela 1: Comparativo entre os trabalhos relacionados.....	44
Tabela 2: Respostas as critérios no modelo UFollower	46
Tabela 3: Nodos Pessoais utilizados na simulação do modelo UFollower	64
Tabela 4: Câmeras utilizadas na simulação do modelo UFollower.....	65
Tabela 5: Veículos utilizados na simulação do modelo UFollower	65
Tabela 6: Amostra de Dados tabelados para o modelo RB	68
Tabela 7: Algoritmos baseados em Naive Bayes	68
Tabela 8: Histórico de Contexto veículo	76
Tabela 9: Câmeras de acesso aberto utilizadas na simulação do modelo.....	77
Tabela 10: Mensagens personalizadas no modelo UFollower	79
Tabela 11: Turistas introduzidos na simulação	81

LISTA DE SIGLAS

API	<i>Application Programming Interface</i>
ASC	<i>Assistive Smart City</i>
ARBSVU	Agente de Recomendação Baseado em Segurança e Vigilância Ubíquas
DAO	<i>Data Access Object</i>
DAER	<i>Departamento Autônomo de Estradas de Rodagem</i>
DLUT	<i>Dalian University of Technology</i>
ESB	<i>Enterprise Service Bus</i>
FIM	<i>Fast Iterative Matching</i>
GPS	<i>Global Positioning System</i>
HAR	<i>Human Activity Recognition</i>
HCI	<i>Human Computer Interface</i>
IaaS	<i>Infrastructure as a Service</i>
ICT	<i>Information and Communication Technology</i>
IDE	<i>Integrated Development Environment</i>
IDS	<i>Intrusion Detection System</i>
IoT	<i>Internet of Things</i>
JSON	<i>Javascript Object Notation</i>
MCS	<i>Mobile Crowdsensing</i>
MVC	<i>Model-View-Controller</i>
M2M	<i>Machine to Machine</i>
OMS	Organização Mundial da Saúde
ONU	Organização das Nações Unidas
PaaS	<i>Platform as a Service</i>
PCD	Pessoa Com Deficiência
PDT	<i>Prometheus Design Tool</i>
P2M	<i>Peer to Machine</i>
P2P	<i>Peer to Peer</i>
RB	Redes Bayesianas
RDBMS	<i>Relational Database Management Systems</i>
REST	<i>Representational State Transfer</i>
RFID	<i>Radio-Frequency Identification</i>
SaaS	<i>Software as a Service</i>
SMA	<i>Single Page Application</i>
SOA	<i>Service-Oriented Architecture</i>

<i>SPA</i>	<i>Single Page Application</i>
<i>TAM</i>	<i>Technical Architecture Modeling</i>
TIC	Tecnologias da Informação e Comunicação
TI	Tecnologia da Informação
u-City	<i>Ubiquitous City</i>
UML	<i>Unified Modeling Language</i>
<i>URL</i>	<i>Uniform Resource Locator</i>
USGS	<i>United States Geological Survey</i>
USP	Universidade de São Paulo
WBAN	<i>Wireless Body Area Network</i>
Wi-Fi	<i>Wireless Fidelity</i>

SUMÁRIO

1 INTRODUÇÃO.....	18
1.1 Motivação.....	14
1.2 Definição do Problema e Questões de Pesquisa.....	16
1.3 Objetivos.....	16
1.4 Metodologia.....	17
1.5 Organização do Trabalho.....	17
2 FUNDAMENTAÇÃO TEÓRICA	19
2.1 Computação Ubíqua.....	20
2.1.1 Computação em nuvem.....	22
2.1.2 Internet das Coisas.....	24
2.1.3 <i>Wearables</i>	25
2.2 Redes Bayesianas	33
2.3 Segurança Ubíqua.....	30
2.4 Classificação das Cidades.....	31
2.5 Cidades Inteligentes.....	32
2.6 Considerações sobre o Capítulo.....	33
3 TRABALHOS RELACIONADOS.....	34
3.1 Critérios para escolha dos trabalhos.....	34
3.2 <i>Privacy and Security Aspects of E-Government in Smart Cities</i>.....	34
3.3 <i>SPATH: Finding the Safest Walking Path in Smart Cities</i>.....	36
3.4 <i>Smart services: A case Study on smarter public safety by a mobile app for University of São Paulo</i>.....	38
3.5 <i>Urban Infrastructure safety system based on mobile crowdsensing</i>.....	39
3.6 <i>Smart surveillance em aplicações recentes no Brasil: um estudo de caso nas cidades de Recife e Curitiba</i>.....	41
3.7 Análise comparativa e contribuição do modelo UFollower.....	43
3.8 Considerações sobre o capítulo.....	46
4 MODELO UFOLLOWER.....	51
4.1 Visão geral do modelo.....	47
4.2 Requisitos do Modelo.....	48
4.3 Arquitetura do modelo UFollower.....	49

	13
4.3.1 Servidor UFollowerServer.....	51
4.3.2 Ontologia do modelo UFollower.....	52
4.3.3 Sistema Multiagente do modelo UFollower.....	53
4.3.4 Aplicativo Móvel e Retaguarda <i>Web (Site)</i>	58
4.4 Considerações sobre o capítulo.....	59
5 ASPECTOS DE AVALIAÇÃO E IMPLEMENTAÇÃO	60
5.1 Implementação do protótipo.....	60
5.1.1 Implementação do Servidor UFollower.....	60
5.1.2 Implementação dos Clientes.....	62
5.2 Metodologia e aspectos de avaliação.....	62
6 CONSIDERAÇÕES FINAIS.....	85
6.1 Contribuições.....	85
6.2 Limitações e Trabalhos Futuros.....	85
REFERÊNCIAS.....	93

1 INTRODUÇÃO

As cidades estão se tornando maiores e mais complexas e enfrentam cada vez mais desafios com o rápido aumento da população urbana em todo o mundo (NAVARATHNA; MALAGI, 2018). Algumas tecnologias podem ser adotadas para agilizar e facilitar as tarefas do cotidiano. Exemplos: *machine learning*, computação ubíqua, tecnologias sem fio e Internet das coisas (IoT do inglês *Internet of Things*), através de sensores, medidores e dispositivos pessoais (PIRO et al., 2014; ICHIFUJI et al., 2016; QAMAR et al., 2019).

Tais tecnologias podem ser utilizadas para acompanhar um indivíduo dentro de um deslocamento urbano, acompanhando-o de tal forma que a coleta de informações possa lhe trazer uma assistência ubíqua e personalizada. Essa assistência voltada à segurança, pode inferir, por exemplo, a iminente entrada de um turista em zona de risco, notificando-o. Por zona de risco, entende-se um local onde a ocorrência de determinados tipos de crimes acontecem com frequência. Pode, ainda, auxiliar na tomada de decisão de uma vítima, ao identificar que um agressor já conhecido, monitorado, está se aproximando de sua residência ou está prestes a passar próximo na rua. Nessa mesma situação, agentes de segurança podem receber a mesma notificação visando uma ação proativa.

Dentre os diversos segmentos que compõem a estrutura de uma cidade, um merece especial atenção: a segurança pública. No Brasil, estudos realizados no ano de 2019, conforme dados do Atlas de Violência IPEA (2019), demonstram que a taxa de homicídios no Brasil aumentou 4,2% de 2016 a 2017, estabelecendo um novo recorde com 31,6 mortes a cada 100 mil pessoas. Em termos absolutos, os assassinatos totalizaram 65.602 em 2017, um aumento de 4,9% em relação a 2016 (LISBOA, 2019).

Em uma cidade grande e moderna, o sistema de segurança não é um módulo único de uma só camada. Em vez disso, consiste em muitas camadas com subsistemas autônomos que são capazes de interagir, reciprocamente, de maneira eficiente e coordenada, horizontal e verticalmente JAMEEL (2019). Segurança e privacidade para soluções de cidades inteligentes foram poucas vezes consideradas uma prioridade pela comunidade das *smart cities*. É necessário incentivar cidades e comunidades inteligentes a abranger a segurança e a privacidade como uma preocupação primária (TOUSLEY; RHEE, 2018).

O Brasil tem despertado para a necessidade de adequar suas cidades e para isso iniciou em 2017 o desenvolvimento de um projeto piloto em algumas cidades brasileiras focado em cidades inteligentes e criativas. É fato que a maioria das cidades não apresenta condições ideais para que a população usufrua de ambientes inteligentes, atrativos e melhores para se viver. Entre as adversidades, citam-se os contratempos de um modelo urbano, que priorizou a utilização de veículos automotores individuais, com a carência de espaços e de equipamentos públicos e comunitários (PIEKAS et al., 2018).

Diante deste cenário, é importante que sejam desenvolvidas ferramentas baseadas na tecnologia da informação e comunicação que auxiliem a população nas atividades do seu cotidiano, levando-se em consideração, primariamente, a segurança assistida do indivíduo.

1.1 Motivação

A necessidade de acompanhar de forma ubíqua o deslocamento de um indivíduo dentro de uma área geográfica preestabelecida pode aumentar a qualidade de sua vida. Pelo fato de

identificar uma pessoa dentro de um determinado perfil nessa área, seja morador ou turista, por exemplo, já é suficiente para lhe fornecer um serviço mais adequado ao seu perfil.

Há também a demanda crescente por aplicações que podem detectar mudanças no afeto ou comportamento humano e sugerir uma ação, sendo a análise do afeto humano cada vez mais utilizada nas áreas de segurança, saúde e comunicação social. Dessa forma, existe a necessidade de extrair esses descritores latentes, dados oriundos de áudio, vídeo ou sensores (LADE et al., 2013).

A acessibilidade é outro ponto que pode ser discutido nesse cenário de acompanhamento ubíquo. Ela tem o objetivo de acompanhar pessoas com necessidades especiais, mobilidade reduzida, deficientes auditivos e visuais. Outro ponto importante discutido por (TELLES; BARBOSA; RIGHI, 2017) é a infraestrutura disponível nas regiões urbanas. Para citar a principais: conectividade, rede de sensores, computação ubíqua, juntamente com os recursos colaborativos, isso é, informações originadas pelos cidadãos, podem oferecer subsídios para a implantação de serviços destinados às pessoas com deficiência (PCDs), promovendo a acessibilidade. A computação ubíqua utilizada e discutida nesta abordagem, envolve computação em nuvem, *wearables*, IoT e Redes Bayesianas.

A vigilância ubíqua de um indivíduo pode também servir como ato ativo na prevenção de crimes. Estratégias de policiamento preditivo estão cada vez mais sendo aprimoradas. Alguns departamentos de polícia têm registrado eventos criminais em seus bancos de dados para compor estratégias melhores de como entender e prever a incidência de crimes. Essa previsão de incidentes criminais ajuda os supervisores a elaborarem a lista de locais predefinido e o tempo de permanência em que cada veículo policial deve ficar no local. Os aplicativos devem suportar múltiplas seleções de métodos estatísticos e de aprendizado de máquinas para criar um ambiente de previsão do crime em diferentes áreas da cidade (JUNIOR et al., 2017).

Um exemplo de crime com alta taxa de ocorrências no Brasil, cuja prevenção pode ser melhorada através do emprego de acompanhamento ubíquo, é o feminicídio. Sabe-se que o Brasil tem a quinta maior taxa de feminicídio do mundo, conforme números da Organização Mundial da Saúde (OMS). O número de assassinatos no país chega a 4,8 casos para cada 100 mil mulheres (ONU, 2016). Com as tecnologias já discutidas relativas à computação ubíqua, relacionadas a *hardwares* externos, como uma tornozeleira eletrônica e câmeras de segurança, por exemplo, pode-se confrontar a posição geográfica de um agressor com a localização de uma vítima, podendo-se evitar um crime e o incremento desta estatística.

O histórico de contextos (VIANNA; BARBOSA; PITTOLI, 2017) poderá ser utilizado para inferir determinadas situações. Uma área geográfica tem referências de recursos, pessoas e informações de contexto associadas a esse local. Essas informações podem ser outro local, uma pessoa ou um recurso. Além disso, pode-se compartilhar essas informações contextuais. O contexto é a temperatura de um local, a frequência cardíaca de uma pessoa ou o nível de combustível de um carro, por exemplo. E o histórico de contexto pode ser utilizado para prevenção de uma determinada situação. Por exemplo, um apenado detido por violência doméstica recebeu progressão de pena e está sendo monitorado através de tornozeleira eletrônica nas ruas. Se a vítima dele, a qual já tenha registrado ocorrência contra este agressor, estiver geograficamente, sem o seu conhecimento, aproximando-se dele (involuntariamente, um caminhando contra o outro) ela poderá receber uma notificação para se afastar do local, como medida de precaução, protegendo-se física e psicologicamente.

Neste trabalho é proposto o modelo UFollower (*Ubiquitous Follower*) para uma cidade inteligente. O modelo UFollower é destinado, de modo geral, aos usuários que necessitam de algum acompanhamento e vigilância ubíquos.

1.2 Definição do Problema e Questões de Pesquisa

A implementação de uma cidade inteligente (SHAHROUR; ALILECHE; ALFURJANI, 2017) visa melhorar a eficiência e segurança, bem como qualidade de vida dos cidadãos. O conceito de *smart city* é baseado na coleta e análise de dados, a qual em diferentes escalas e em diferentes contextos permite melhorar a nossa compreensão do funcionamento de sistemas urbanos e melhorar tanto a sua segurança como eficiência. A segurança e privacidade de indivíduos em cidades inteligentes geralmente não são consideradas as principais prioridades para a comunidade. Normalmente se tem soluções e projetos pontuais que não possuem um modelo operacional além de sua implantação inicial (TOUSLEY; RHEE, 2018).

Neste sentido, um acompanhamento ubíquo do indivíduo, tendo como premissa fundamental a assistência à segurança e de forma primária. Levando-se em consideração o fato de que o estado da arte garante a concepção de um modelo, que possa se utilizar de informações já existentes, dentro de um contexto geográfico bem definido, fazendo uso de imagens, sensores, perfis de usuário e outras entradas que atendem à computação ubíqua, torna-se possível a criação de um modelo. Neste cenário, surge a seguinte questão de pesquisa: como a computação ubíqua pode ser aplicada na criação de um modelo computacional para uma cidade inteligente que promova a segurança dos indivíduos?

1.3 Objetivos

Este trabalho tem como objetivo geral a criação de um modelo computacional, para suporte às cidades inteligentes que promova o acompanhamento ubíquo do indivíduo, levando em consideração questões como perfil de usuário, geolocalização, inferências bayesianas e a vigilância ubíqua.

Para tanto, serão explorados principalmente os trabalhos na área de segurança em cidades inteligentes. Para atingir este objetivo geral são destacados os seguintes objetivos específicos:

- Aprofundar os conceitos relacionados aos assuntos propostos, ou seja, conceitos relevantes sobre segurança em cidades inteligentes, acessibilidade e histórico de contexto. Na computação ubíqua, os temas de computação em nuvem, Internet das Coisas, histórico de contexto e *wearables* são estratégicos, pois serão a base tecnológica do modelo;
- Realizar um estudo sobre os trabalhos relacionados à segurança em cidades inteligentes, para identificar as funcionalidades já existentes e, também, as demandas não atendidas;
- Elaborar um modelo para aplicação da computação ubíqua na criação de uma cidade inteligentes assistiva;
- Desenvolver um protótipo do modelo, visando atender as demandas identificadas nos trabalhos relacionados;
- Avaliar o modelo com o protótipo desenvolvido, utilizando uma abordagem quantitativa, aplicando o método definido como pesquisa experimental.

1.4 Metodologia

Para um melhor entendimento do contexto atual acerca das cidades inteligentes, considerando o nível de tecnologia presente e o entendimento das necessidades apresentadas pelos indivíduos, foi realizada uma análise bibliográfica a respeito do tema. De forma a complementar às propostas identificadas na literatura, percebeu-se a necessidade de criar um modelo complementar aos mecanismos presentes nas cidades inteligentes estudadas, o qual possa adotar tecnologias para proporcionar segurança nestas cidades, tornando-as assistivas, levando em consideração o perfil de usuário e vigilância ubíqua.

Uma vez percebidas as possibilidades de melhoria, verificou-se quais recursos a segurança, dentro do contexto de uma cidade inteligente, poderiam ser aplicados com o objetivo de atender de forma colaborativa e integrada aos cidadãos. Colaborativa com o intuito de que os próprios cidadãos atuassem, com a devida limitação legal, como agentes, e garantindo a segurança do outro. Integrada, pois o perfil pré-cadastrado desses indivíduos serviria para farmácias, hospitais e turismo. O histórico de contexto coletaria, com a ciência do indivíduo, informações temporais para que sugestões pudessem lhe ser prestadas.

De forma a anteceder o desenvolvimento do modelo UFollower e seus componentes, faz-se necessário a realização de uma análise dos possíveis cenários de sua aplicabilidade nas mais diversas situações do dia a dia dos indivíduos envolvidos. Dessa forma, serão analisadas as demandas atuais não atendidas no cotidiano no que se refere à segurança pública e histórico de contexto. Por exemplo, a inferência de um evento que culmine em um fato criminal e uma ação de emergência acionada por um indivíduo em perigo enviando as informações para uma retaguarda da segurança pública. Há também a sugestão de mensagem informativa alertando um turista de perigo tendo como base a sua localização geográfica e a incidência de crimes no local.

Tendo os cenários mapeados, criou-se um modelo para a segurança pública voltado para cidades inteligentes. Na primeira etapa do desenvolvimento do modelo foram identificadas as funcionalidades e classificação dessas cidades inteligentes, levando-se em consideração o nível de tecnologia usado. Na segunda etapa, os trabalhos relacionados foram analisados com o intuito de estabelecer quais partes deveriam ser projetadas. A terceira etapa, consistiu no desenvolvimento do modelo computacional, para suporte à segurança pública, visando utilizar o perfil dos usuários e *smart surveillance* nas cidades inteligentes, tornando-as assistivas. Na quarta e última etapa, objetivou-se a avaliação do modelo computacional criado, simulado, onde agentes foram inseridos no modelo.

1.5 Organização do Trabalho

Esta dissertação está dividida em seis capítulos. O capítulo 2, discute a fundamentação teórica e tecnologias que servem como base para o desenvolvimento do modelo UFollower, entre as tecnologias, destacam-se computação ubíqua, computação em nuvem, IoT *wearables*, histórico de contextos e Redes Bayesianas. A fundamentação teórica discute também, a segurança ubíqua e a classificação das cidades. No capítulo 3 são apresentados os trabalhos relacionados e um estudo comparativo com características, funcionalidades e limitações, bem como a contribuição do modelo UFollower. O modelo proposto para cidade inteligente assistiva, com sua arquitetura é apresentado no capítulo 4. No capítulo 5 são apresentados o protótipo do modelo e a descrição da metodologia para realização de experimentos de avaliação

do modelo UFollower. Por fim, o capítulo 6 contém as considerações finais, contribuições esperadas, cronograma com as próximas etapas a serem seguidas e produção de artigos.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma descrição dos conceitos e tecnologias para elaboração desta proposta de dissertação. A Figura 1 mostra um esquema dos principais conceitos adotados, para desenvolvimento do modelo UFollower.

Figura 1: Conceitos do modelo UFollower



Fonte: Elaborado pelo Autor.

A Figura 1 indica que a junção das tecnologias da computação ubíqua com as iniciativas de segurança resulta na segurança ubíqua. O modelo UFollower colabora para a constituição deste conceito de cidade, uma vez que adota tecnologias da computação ubíqua, juntamente com outros recursos descritos neste capítulo. Com relação à segurança, o modelo UFollower atende uma série de normas, oferecendo suporte a vítimas de feminicídio e turistas, para que estes possam utilizar os recursos em seu cotidiano e realizarem suas atividades com mais autonomia.

Para modelar o suporte à segurança, voltado aos ambientes das cidades inteligentes, é necessária a exploração das tecnologias envolvidas, com uma visão sobre segurança e um estudo sobre os cenários, que constituem as cidades inteligentes. Desta forma o presente capítulo busca descrever conceitos de tecnologias para desenvolvimento do modelo UFollower, segurança ubíqua e classificação das cidades.

O texto está organizado em cinco seções. A primeira seção, descreve as tecnologias necessárias para o modelo UFollower. São discutidas as tecnologias de computação ubíqua abordando computação em nuvem, *wearables*, IoT e histórico de contextos. A segunda seção deste capítulo, versa sobre segurança ubíqua. A terceira seção trata da classificação das cidades, com relação ao nível de tecnologia adotada. A quarta seção dedica-se a especificar, detalhadamente, as cidades que são classificadas como *smart city*, pois trata-se da classificação foco para o modelo UFollower. No restante do trabalho, será adotado o termo cidade inteligente,

para se referir ao paradigma *smart city*. Por fim, a quinta seção relata as considerações sobre o capítulo.

Com relação à segurança, o modelo UFollower atende uma série de itens, oferecendo acompanhamento ubíquo assistido para que o indivíduo possa se deslocar pelo perímetro de abrangência do modelo com segurança. Para modelar o suporte à segurança voltado aos ambientes das cidades inteligentes, é necessária a exploração das tecnologias envolvidas, com uma visão sobre segurança e um estudo sobre os cenários, que constituem as cidades inteligentes. Dessa forma, o presente capítulo descreve os conceitos de tecnologias para desenvolvimento do modelo UFollower, segurança ubíqua e classificação das cidades.

O presente capítulo aborda o tema da computação ubíqua, computação em nuvem, Internet das coisas, *wearables* e histórico de contextos, bem como a aplicabilidade da computação ubíqua na área da segurança. Além disso, se propõe a apresentar os principais conceitos relacionados a RBs. No final do capítulo é abordado o conceito de segurança ubíqua, classificação das cidades e cidades inteligentes.

2.1 Computação Ubíqua

Nesta seção, uma abordagem sobre as tecnologias de ICT é apresentada, tecnologias estas que servirão de base para desenvolvimento do modelo UFollower. Serão trazidos conceitos da computação ubíqua, *wearables*, modalidades da computação em nuvem e uma descrição das aplicações de IoT.

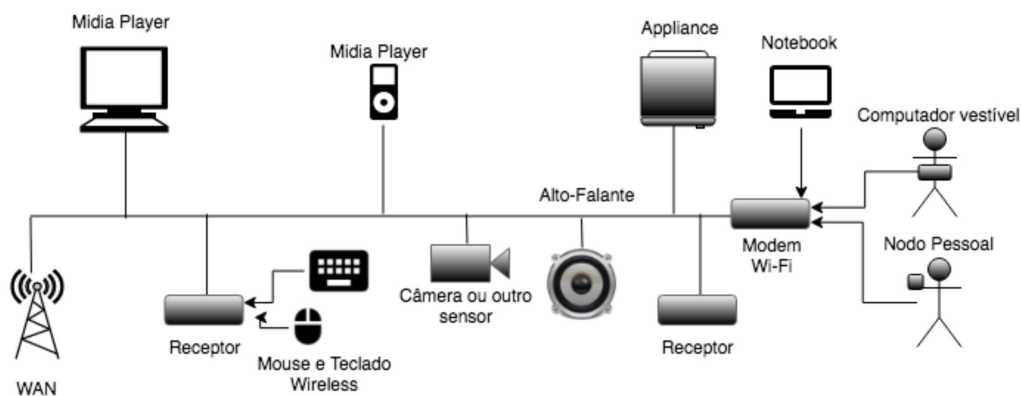
O progresso das tecnologias de informação e comunicação, oportunizaram o estabelecimento de disposições apontadas por (WEISER 1991), onde, valendo-se de uma concepção futurista, percebeu que a computação seria despercebida às possibilidades advindas dos recursos computacionais seriam utilizados, por parte dos usuários, sem a real consciência de sua utilização. Esta tendência, é um dos conceitos centrais da computação ubíqua, onde o uso de recursos ou conjunto de recursos ocorre de forma imperceptível.

Uma concepção pertinente para a definição de recursos na computação ubíqua, é a sensibilidade ao contexto (DEY; ABOWD; SALBER, 2001; CHEN; KOTZ, 2000; ABOWD et al., 1999). O contexto pode ser entendido como qualquer informação considerável, que caracterize pessoas, objetos ou lugares. O acesso a essas informações pode se dar sem a necessidade explícita do usuário, ou seja, pode-se alcançá-las automaticamente. Utilizando-se desse contexto para realizar inferências ou ações que visam melhorar a experiência dos usuários, a sensibilidade ao contexto é aplicada. Condições para que a utilização de *software* fosse capaz de oportunizar a criação de ambientes inteligentes foram proposta por (DEY; ABOWD; SALBER, 2000; DEY, 2001; NIEMELÄ; LATVAKOSKI, 2004). Exemplos demonstram que sistemas computacionais podem atuar, de forma espontânea, na prestação de melhores serviços, tendo como objetivo os usuários e seu cotidiano.

Estas realizações oportunizaram que os desenvolvedores pudessem entender melhor e entrar no desenvolvimento de ambientes inteligentes, por exemplo, para uma cidade inteligente. Para (SHIN et al., 2008), com a difusão da computação ubíqua, muitas instalações nas cidades que estão intimamente relacionadas com a nossa vida diária estão sendo combinadas com tecnologias. Como resultado, serviços de TI são integrados, formando uma *u-City* (cidade ubíqua). Uma *u-City* pode ser definida como uma cidade avançada que é capaz de desempenhar o papel de proporcionar maior conveniência para os cidadãos, melhorando a qualidade de vida e proporcionando a gestão sistemática da cidade.

De forma conceitual, pode-se definir a computação ubíqua pela representação da junção da computação móvel com a computação pervasiva. Na Figura 2, SHAFER (2000) apresenta alguns dos elementos que representam a computação ubíqua. É possível notar a presença da computação em nuvem, dispositivos de IoT, objetos *wearables*, redes *wireless*, isso para citar alguns. A compreensão de todos esses itens torna a computação ubíqua compreensível, pois nela se encontra uma situação onde existe alto nível de mobilidade e de computação pervasiva.

Figura 2: Elementos representativos da computação ubíqua

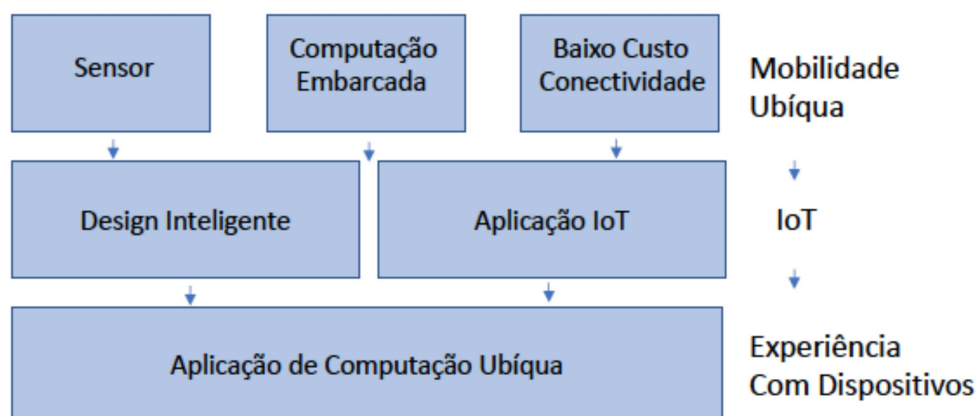


Fonte: adaptado de SHAFER (2000).

A compreensão do conceito computação ubíqua permite o entendimento de que ela trouxe diversos serviços os quais ficam à disposição dos usuários e muitas condições colaboraram para a expansão desta situação. Para citar as que mais se destacam: o custo dos dispositivos diminuiu, há vasto conjunto de desenvolvedores, de forma geral, é fácil utilizar e o retorno dos benefícios oferecidos é facilmente percebido. Entre os benefícios, destacam-se a distribuição de conteúdo, saúde (VIANNA et al., 2012), educação (BARBOSA et al., 2008) e segurança (FERRAZ; ANDRÉ; FERRAZ, 2014).

A Figura 3 apresenta alguns conceitos presentes na computação ubíqua. Segundo ROUSE (2016), a computação ubíqua tem a tendência crescente de incorporar recursos computacionais em objetos do cotidiano para fazê-los se comunicar e executar tarefas úteis de maneira eficaz, minimizando a necessidade do usuário final de interagir com os computadores. Dispositivos de computação ubíqua estão conectados à rede e estão constantemente disponíveis. Desta forma, torna-se possível identificar dados vindos dos mais diversos objetos aos quais considerarmos. Por exemplo, se o foco for uma pessoa, é possível saber se ela está se movimentando, se o ambiente onde ela está esquentou ou ficou frio, se há ruído no local, até mesmo se ela está sozinha ou em grupo. Se o enfoque for um objeto, há como inferir o seu estado atual, se está sendo usado ou não, se está operante ou em manutenção, entre outras informações.

Figura 3: Elementos e características da computação ubíqua



Fonte: Adaptado de ROUSE (2016).

A Figura 3 apresenta a importância do papel de sensores, computação embarcada e de baixo custo de conectividade em aplicações de Internet das coisas, tudo isso em um contexto único de computação ubíqua. Em um fluxo de leitura de cima para baixo, temos na camada de cima a mobilidade ubíqua, na camada do meio a parte IoT e na última camada, a experiência alcançada pelas aplicações ubíquas.

Além dos conceitos já citados, a computação ubíqua tratada neste trabalho, considera ainda a junção das quadros tecnologias: computação em nuvem, IoT, *wearables* que serão apresentadas a seguir juntamente com o histórico de contextos, parte fundamental para a concepção do UFollower.

2.1.1 Computação em nuvem

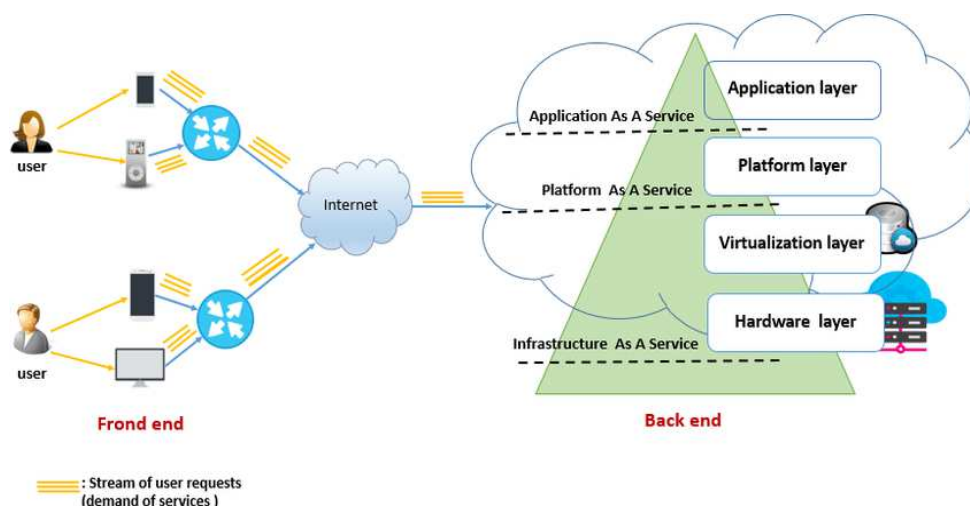
A computação em nuvem tem o potencial de deixar o *software* ainda mais atraente como serviço, de forma a não exigir grandes gastos de capital em *hardware* para serem implantados ou de despesas humanas para operá-lo. Este conceito segundo ARMBRUST et al. (2010) se chama demanda *online*. Nesta definição, pode-se atribuir alguns tipos de serviço de *cloud*, entre elas, destacam-se *software* como serviço (SaaS), que trata da maneira de transportar aplicativos como serviço na Internet, ao invés de instalar o *software* em seu computador, o usuário pode simplesmente acessá-lo via Internet, Plataforma como serviço (PaaS), onde um ambiente ou plataforma de desenvolvimento é fornecido aos consumidores como um serviço, no qual o usuário pode implantar seu próprio *software* e codificação, Infraestrutura como serviço (IaaS), nesta modalidade muitos recursos de computação são fornecidos na forma de dispositivos de armazenamento, rede, sistema operacional, *hardware* e armazenamento sob demanda. Os usuários de IaaS podem acessar os serviços usando uma rede de área ampla, como a Internet. Por exemplo, um usuário pode criar máquinas virtuais efetuando *login* na plataforma IaaS (SRIVASTAVA; KHAN, 2018).

Nuvem pública, trata-se de uma nuvem como um serviço de computação fornecido por terceiros, como na Internet. Esses serviços estão disponíveis para qualquer usuário que queira usá-los e eles não precisam pagar apenas pelos serviços consumidos. Nuvem privada é onde os serviços de computação fornecidos pela Internet ou rede privada são oferecidos apenas aos

usuários selecionados no lugar de pessoas comuns. Uma maior segurança e privacidade são delegadas por nuvens privadas através do *firewall* e hospedagem interna. A nuvem híbrida traz a combinação de nuvem pública e nuvem privada. Na nuvem híbrida, cada nuvem pode ser gerenciada independentemente, mas dados e aplicativos podem ser compartilhados entre as nuvens no formato híbrido (GARRISON et al., 2012).

Na Figura 4 é apresentada a arquitetura da computação em nuvem em camadas (AGARWAL, 2019). A arquitetura de computação em nuvem pode ser classificada em duas divisões; *front-end* e *back-end*. O *back-end* da *Cloud Architecture* fornece segurança de dados para os consumidores da nuvem e também possui um mecanismo de controle de tráfego que gerencia a carga no servidor. O servidor também fornece o *middleware* que ajuda a conectar dispositivos e se comunicar. O *front-end* da *Cloud* consiste em plataformas chamadas *Cloud Clients*, que incluem servidores, *thin* e *fat client*, *tablets* e dispositivos móveis. A interação é feita através do *middleware* ou através de sessões virtuais do navegador da *Web*.

Figura 4: Representação da arquitetura de computação em nuvem em camadas *Front End* e *Back End*



Fonte: AGARWAL (2019).

Com o intuito de possibilitar o atendimento das mais diversas demandas, tanto de desenvolvedores como de empresas, a computação em nuvem surge como alternativa, uma vez que possui funcionalidade de interoperabilidade, elasticidade, descentralização, customização e segurança (MELL; GRANCE, 2011).

Alguns modelos já propostos, visavam atender regiões limitadas (geograficamente e computacionalmente), não sendo a elasticidade prevista. É fundamental que se pense em elasticidade quando o sistema tende a ser utilizado por muitos usuários, ainda mais em situações onde usuários lotados em locais distantes fisicamente. Servidores distintos, necessitam interagir, como se estivessem próximos, sem ficarem perceptíveis a utilização dos demais usuários. Prevendo a possibilidade de utilização *off-line*, seja por falta de conectividade ou pelas informações não estarem disponíveis, módulos de operação e simulação estarão disponíveis.

Os custos de equipamentos, infraestrutura e da própria elasticidade são discutidos por (CABALLER et al. 2013), onde são indicados diferentes formas de implementação para atender a elasticidade de modo a contemplar uma demanda variada, por exemplo, com muitos usuários

utilizando dados e conexões, concomitantemente. Algumas das formas discutidas vão desde o tipo de *software*, componentes de aplicação e linguagem de desenvolvimento utilizada.

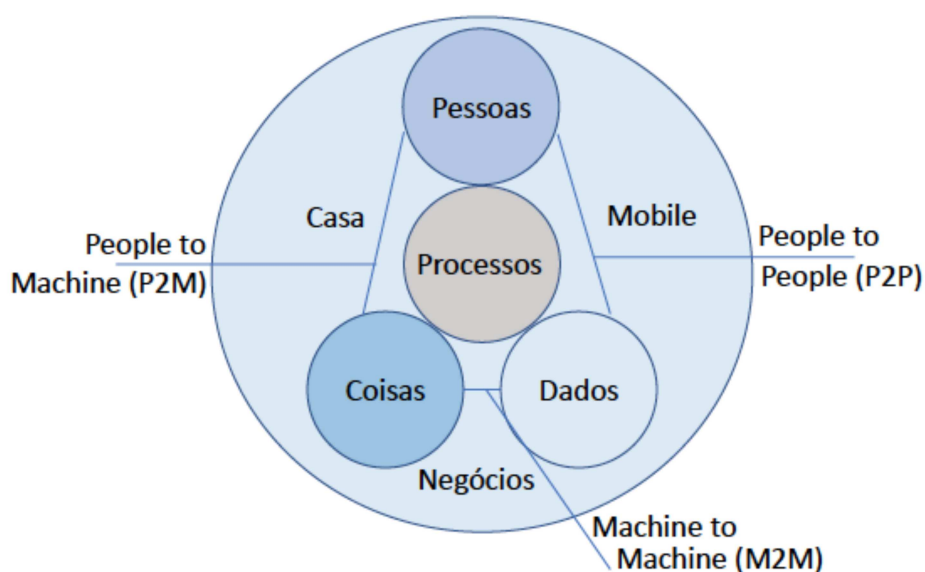
2.1.2 Internet das Coisas

Em nossa vida cotidiana, incluindo nossas casas e locais de trabalho, *gadgets* tornaram-se uma parte imperativa da nossa vida. Estamos caminhando para conectividade a qualquer hora e em qualquer lugar, conectados a qualquer coisa. Com o surgimento de tecnologias como a IoT, estamos em um mundo conectado às coisas ao nosso redor, onde as coisas fazem parte da Internet, combinando perfeitamente o mundo real com o mundo digital (DATTA; SHARMA, 2017).

Para FOX, DONNELLAN, DOUMEN (2019), a IoT está evoluindo rapidamente com uma ampla gama de tecnologias. Da perspectiva de sistema, uma solução completa de IoT pode ser separada em três categorias: o coletor de dados, um método de comunicação e a plataforma em nuvem como serviço. O coletor de dados está relacionado ao dispositivo do sistema incorporado na origem do aplicativo, o método de comunicação refere-se ao protocolo de rede usado para enviar ou receber os dados e o serviço em nuvem se relaciona com a infraestrutura usada para armazenar e processar os dados coletados.

Para MIRAZ et al. (2015) a IoT pode ser entendida como quatro pilares compostos de pessoas, dados, processos e “coisas”. Esse conceito é utilizado no modelo U-Follower. As pessoas vão se deslocar no modelo de cidade inteligente o qual será elaborado e poderão tanto consumir quanto fornecer dados para as “coisas” do modelo. Coisas estas que poderão ser: um objeto *wearable*, uma câmera de segurança, um cadastro em uma farmácia ou um boletim de ocorrência em uma delegacia. Da mesma forma, uma cidade inteligente deverá ser projetada para atender a essas demandas e processos deverão ser criados para gerir da melhor forma o modelo como um todo. Na Figura 5 são apresentados os pilares da Internet das coisas.

Figura 5: Pilares da Internet das coisas



Fonte: Adaptado de MIRAZ et al. (2015).

Na Figura 5 são apresentados os pilares da Internet das Coisas, sendo estes: os dados, as coisas que os geram e as pessoas que os consomem, tudo isso, dentro de um processo definido. Esse processo, no exemplo, é obtido através de ligações do tipo: pessoa para pessoa (P2P), pessoa para máquina (P2M) e *machine to machine* (M2M).

Uma das possibilidades fornecidas pelo uso de IoT é a obtenção de dados provindos de ambientes, sendo coletados no local de onde o dado está, com um detalhamento específico, por exemplo, a localização de um objeto em algum lugar, o controle de um objeto em um ambiente específico. Além desse detalhamento *in loco*, a IoT traz a possibilidade de adicionar semântica aos objetos do mundo real (LÓPEZ et al., 2012).

A Internet das coisas será utilizada no modelo através da coleta e análise de dados emitidos por sensores, principalmente em dispositivos vestíveis, tais como: pulseiras, relógios inteligentes e tornozeleiras. Também poderão ser utilizados sensores disponíveis no contexto, como por exemplo, um sensor acústico identificador de disparo.

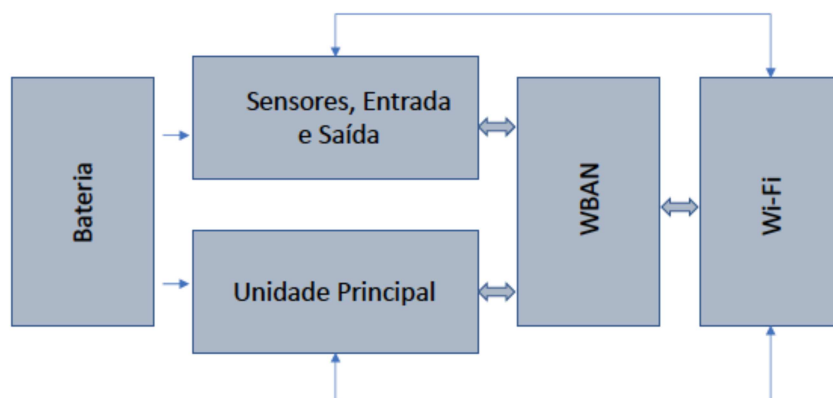
2.1.3 Wearables

Os objetos vestíveis fazem cada vez mais parte da vida das pessoas. Há diversos modelos, formas, objetivos e preços para adquiri-los. Para JHAJHARIA (2014) as tecnologias vestíveis oferecem muitas oportunidades que estimulam os pensamentos e a imaginação das pessoas de todos os campos. Nesta era da tecnologia, a dependência de computadores e outras interfaces exige que os *wearables* se tornassem onipresentes. Esse requisito abriu caminho para o desenvolvimento de diversas tecnologias e objetos que podem auxiliar nas atividades pessoais, aumentando a experiência dos usuários com o mundo da tecnologia.

As muitas oportunidades oferecidas pela computação vestível provocaram a imaginação de *designers* e pesquisadores em uma ampla variedade de campos. A invenção de computadores e interfaces pequenos o suficiente para serem usados no corpo humano, inspirou a criação de dispositivos e aplicativos que possam auxiliar nas atividades profissionais e pessoais especializadas, além de ajudarem a vida cotidiana no mundo moderno (RANDELL, 2005).

A computação vestível é um novo campo de pesquisa que sofreu grandes melhorias na última década e parece que se tornará ainda mais importante no futuro próximo. Os esforços na computação vestível tornaram os dispositivos mais funcionais e atraentes para os usuários. Com *displays* inovadores, com uma maior variedade de modalidades de entrada e tamanhos mais compactos, Um dispositivo de computação vestível típico é mostrado no bloco diagrama apresentado na Figura 6.

Figura 6: Diagrama de blocos de computador vestível típico



Fonte: Adaptado de SMAILAGIC; SIEWIOREK (2015).

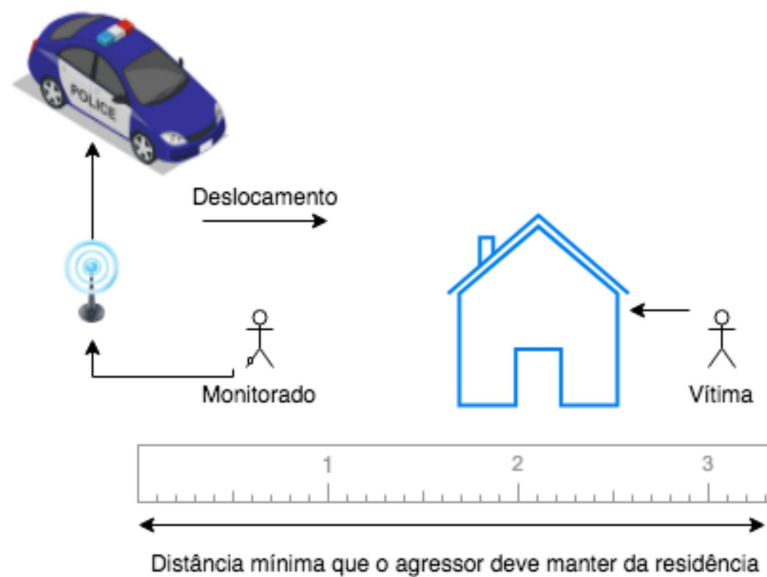
A Figura 6 apresenta os principais componentes de um dispositivo vestível, entre eles: uma bateria, um *human computer interface* (HCI), como sensores e dispositivos de entrada/saída e uma rede de comunicação como *Wireless Body Area Network* (WBAN) e rede *wireless fidelity* (Wi-Fi). A comunicação entre os componentes é direcional, representada na imagem pelas setas.

Com o objetivo de fazer uso de objetos *wearables* reais para uma efetiva tomada de decisão, o modelo UFollower utilizará, por exemplo, dados simulados provenientes de tornozeleiras eletrônicas. O uso das tonozeleiras funciona como uma progressão de pena, uma alternativa à prisão fechada. Segundo SMAILAGIC e SIEWIORK, (2015) nos últimos 20 anos, o uso do monitoramento eletrônico de infratores aumentou muito, reintegrando criminosos de maneira mais eficaz na sociedade. O reintegrado passa por medidas de supervisão necessárias para evitar que ele corrompa o sistema e caso haja o uso irresponsável, ele será penalizado.

Para WILLIAMS e WEATHERBURN, (2019) o monitoramento eletrônico reduz a reincidência em 16% em 2 anos em comparação com cumprimento uma sentença em uma prisão. Para infratores com menos de 30 anos, a redução é de 43%, com reduções significativas na reincidência persistindo por 8 anos. Além disso, há uma economia significativa de dinheiro para cada infrator monitorado, comparando-se com aqueles que não ganham a progressão de pena.

Como pode ser verificado, há resultados positivos no uso de tornozeleiras eletrônicas, no entanto, há de se preocupar com os percentuais reincidentes, aqueles que mesmo desfrutando do benefício da tornozeleira, tornam a cometer alguma infração. É com este objetivo que o UFollower terá entre os seus agentes, um que se encarregue de monitorar este evento. A Figura 7 mostra um cenário onde o monitoramento através de tornozeleira eletrônica é utilizado pelo modelo. Para cada ator monitorado que infrinja a distância mínima que lhe foi ajuizada, deslocando-se em direção à residência de uma vítima, sua tornozeleira eletrônica emitirá um sinal para a viatura policial, outro agente no modelo, mais perto do local. Este sinal deverá notificar com informações do tipo: nome do monitorado, nome da vítima, motivo da notificação, coordenadas do monitorado e distância até ele.

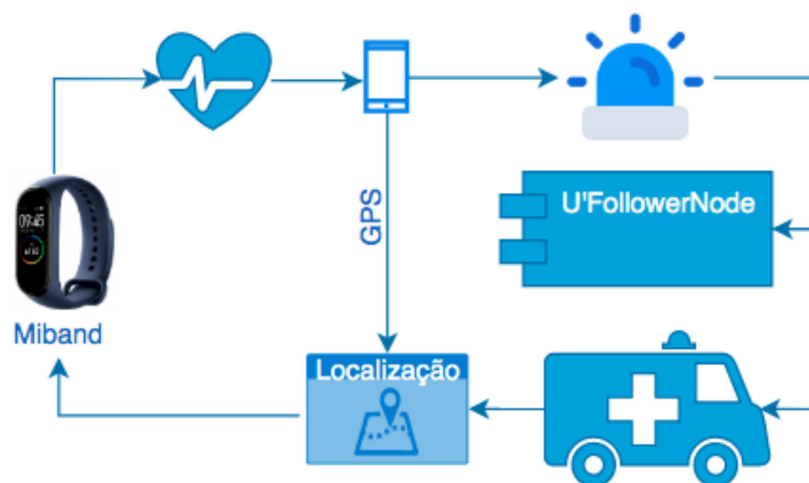
Figura 7: Monitoramento por tornozeleira eletrônica no UFollower



Fonte: Elaborado pelo Autor.

Conforme ilustrado na Figura 8, um outro cenário que pode ser verificado é a utilização do dispositivo vestível fazendo a leitura dos batimentos cardíacos com o propósito de acionar uma emergência médica se necessário. Com a *Miband* é possível verificar o monitoramento 24/7 da frequência cardíaca em tempo real, registrando os dados de saúde e gerando alertas sobre ociosidade. É garantido um intervalo mínimo de 20 dias com um única carga (XIAOMI, 2020).

Figura 8- *Wearable*, Nodo Pessoal e UFollowerNode monitorando batimentos cardíacos



Fonte: Elaborado pelo Autor.

Para SAINATH (2018) a *Miband* pode ser utilizada para a coleta de dados usando um aplicativo chamado *MiFit*, que pode ser baixado do *site* de tecnologias da Xiaomi.

2.1.4 Histórico de Contextos

A sensibilidade ao contexto vem à tona como uma fase do processo evolutivo e está entre os conceitos objetivados por WEISER (1991), quando o mesmo criou o termo computação ubíqua. Para um melhor entendimento é necessário definirmos o que é um contexto. Um dos aspectos relevantes na computação ubíqua é o entendimento do contexto em que se está inserido, ou seja, ser sensível ao contexto (DEY; ABOWD; SALBER, 2001). O termo contexto tem sido estudado em diferentes áreas da Ciência da Computação. Sistemas sensíveis ao contexto têm alta demanda em áreas como ambientes inteligentes, computação pervasiva e ubíqua e inteligência ambiental. Esses sistemas coletam dados e adaptam o comportamento do usuário, usando informações de contexto como: contexto físico, contexto computacional e contexto de tarefas do usuário. Portanto, o termo relativo à computação ubíqua (*Context-Aware Computing*) refere-se, basicamente, a ideia de que dispositivos computacionais podem perceber e reagir ao ambiente físico onde estão implantados (GANDODHAR, 2018).

A computação sensível ao contexto, juntamente com *big data*, pode ilustrar oportunidades e oferecer a melhor escolha para problemas relacionados à áreas urbanas, por exemplo, estruturas de transporte, correspondência, serviços humanos e segurança podem ser beneficiados por essas escolhas (ROOPA, 2018). Os *smartphones* já atuam como uma plataforma promissora para o monitoramento contínuo do comportamento humano (HAR do inglês *Human Activity Recognition*) e com eles é possível capturar os padrões de comportamento das pessoas, dada a variabilidade de configurações e ambientes. O histórico de contexto sendo armazenados ao longo de período e devidamente tratado, pode oferecer um conjunto significativo de possibilidades para reforçar os serviços oferecidos por alguns sistemas de computadores.

Ao realizar a comparação de predição de contextos com determinados tipos de contextos já existentes traz como benefício a detecção de possíveis irregularidades e, com isso, auxilia na identificação de falhas. Há também o benefício da proatividade, uma vez que permite entregar ao usuário interações que atendam melhor às suas expectativas. SMAILAGIC et al. (2020) têm explorado uma série de classificadores *de machine learning* para validar a eficiência do uso de *smartphones* com esse propósito. Segundo os autores, o reconhecimento automático da vida diária humana é muito significativo para aplicações abrangentes relacionadas à robótica, vigilância inteligente, exploração de vídeo na *Web*, e segurança rodoviária (SMAILAGIC, 2020).

RAKITA (2018) propôs um detector de contexto que fornece aos seus assinantes informações sobre mudanças no ambiente, com base no que o assinante definiu como seu interesse. Foi implementado um aplicativo baseado em *Android* para *smartphones*, com o nome “*Active Minutes*”, que fornece ao usuário uma visão personalizada de suas reuniões e tarefas dentro de sua organização. E então o contexto possível é predito utilizando aprendizado de máquina, regras e ontologia.

ZHOU et al. (2019) têm explorado histórico de contextos para a utilização de veículos não tripulados. Para tanto, desenvolveram um novo algoritmo de aprendizado *on-line* utilizando uma estrutura baseada em árvore com reconhecimento de contexto e método de partição ativa para o espaço de contexto.

JOHANNES et al. (2018) têm explorado através do uso de Rede Neural Convolutacional a extração símbolos de assentamentos humanos em mapas topográficos históricos do USGS (*United States Geological Survey*) usando dados de edifícios contemporâneos como a camada espacial contextual. A presença de um edifício na camada contemporânea indica uma alta probabilidade de assertividade de como era o local antes da construção do edifício.

Considerando o ramos das pesquisas, a predição de contextos pode ser aplicada a diversos contextos (VOIGTMANN et al., 2013). Sobretudo, pode ser utilizada para promover a disponibilidade de serviços computacionais para um grande número de usuários, destinando-se às mais diversas áreas, entre elas, a segurança pública.

2.2 Redes Bayesianas

Redes Bayesianas (RB) são modelos probabilísticos que permitem lidar de forma rigorosa com a representação de conhecimentos em domínios onde existe incerteza, ou seja, são modelos probabilísticos através dos quais se pode obter informações sobre a probabilidade de certos eventos acontecerem (VIER; GLUZ; JAQUES, 2015). O nome Redes Bayesianas deriva da utilização da fórmula matemática para o cálculo de probabilidades estabelecido por Thomas Bayes em 1763.

Para ZHANG (2019) a abordagem bayesiana permite estimar e classificar as probabilidades de classificação. A modelagem de rede bayesiana é uma abordagem que permite a análise multivariada e é uma ferramenta que possibilita a representação do conhecimento e do raciocínio sob a incerteza. Os modelos de RB podem ser construídos com base em conhecimentos especializados e também podem ser aprendidos automaticamente a partir dos dados existentes (ONISKO et al., 2019).

Uma maneira de implementar as RB, utilizando o aprendizado de máquina e envolvendo diferentes algoritmos para a realização de inferência estatística é através de bibliotecas que fazem a abstração de modelos. O *Scikit-learn* é uma biblioteca de aprendizado de máquina de código aberto que suporta aprendizado supervisionado e não supervisionado. Ele também fornece várias ferramentas para o ajuste de modelo, pré-processamento de dados, seleção e avaliação de modelo e muitos outros utilitários (SCIKIT, 2020).

Segundo HAO (2019), o Scikit-learn é um pacote de aprendizado de máquina na linguagem de programação Python que é amplamente usado em ciência de dados. O pacote Scikit-learn inclui implementações de uma lista abrangente de métodos de aprendizado de máquina sob convenções unificadas de dados e procedimentos de modelagem, tornando-o um *kit* de ferramentas conveniente para fins estatísticos e comportamentais.

O pacote Scikit-learn contempla algoritmos baseados em RB, entre eles estão: Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes, Bernoulli Naive Bayes e Categorical Naive Bayes. Este trabalho implementará todos para verificar qual atinge a melhor inferência para o crime de feminicídio, tendo como *dataset* gerado a partir da ferramenta Mockaroo. O Mockaroo é uma ferramenta comercial e fácil uso para geração realista de dados, que não requer programação. Uma *interface* simples embutida e uma extensa biblioteca de domínios permitem usuários gerem rapidamente dados realistas com facilidade. O Mockaroo fornece suporte para customizações (MANNINO; ABOUZIED, 2019). Testar com dados realísticos tornará seu aplicativo mais robusto, pois você detectará erros que provavelmente ocorrerão na produção antes do dia do lançamento (MOCKAROO, 2020).

Inferência em RBs diz respeito à tarefa de computar a distribuição de probabilidades a posteriori para um conjunto de variáveis de consulta. É utilizado, basicamente, quando se precisa atualizar ou inferir uma probabilidade de determinado evento. O processo de inferência utiliza-se de informações já existentes na própria rede, porém, para se calcular as probabilidades, é requerida a avaliação de todos os nós pertencentes a rede. Sendo assim, um dos principais problemas observados quando fala-se em inferência em RBs reside no tempo elevado necessário para se realizar uma busca exata (PITTOLI, 2015).

Segundo KOLLER e FRIEDMAN (2009), há três diferentes tipos de algoritmos voltados para a inferência de Redes Bayesianas:

- a) Exatos: quando há um problema de alta complexidade para redes formadas por alto número de nós. Por exemplo, algoritmos que utilizam essa abordagem: Eliminação de Variáveis, Enumeração, Árvore de Junção (*Junction Tree*).
- b) Aproximados: existem com o intenção de diminuir o tempo de execução, comparando-os com os algoritmos de inferência exata. Aqui existe a aproximação das distribuições conjuntas de variáveis da rede. Tais instâncias são denominadas amostras, que representam parte das probabilidades do problema. Esses algoritmos são também chamados de algoritmos de Monte Carlo, pois fornecem respostas aproximadas, cuja exatidão dependerá do número de amostras geradas.
- c) Simbólicos: Os algoritmos de inferência anteriormente descritos necessitam que a função de probabilidade conjunta do modelo seja especificada numericamente, atribuindo-se valores numéricos a todos os parâmetros. Porém, nem sempre a especificação numérica desses parâmetros é possível. Sendo assim, substitui-se os métodos numéricos por métodos simbólicos, capazes de lidar com os parâmetros sem precisar lhes atribuir valor. Normalmente, os métodos de inferência simbólica levam a soluções que se expressam como funções dos parâmetros. As respostas para as questões gerais podem ser dadas na forma simbólica em função dos parâmetros e as perguntas específicas podem ser obtidas substituindo-se os valores dos parâmetros na solução simbólica, não necessitando refazer a propagação.

2.3 Segurança Ubíqua

Na segurança ubíqua existem trabalhos que auxiliam no entendimento de quais são os recursos da segurança ubíqua em uma cidade inteligente. Para IJAZ et al. (2016) cidades inteligentes combinam computação ubíqua e gerenciamento urbano, e a segurança ubíqua está inserida neste contexto, utilizando-se desde plataformas de sensores distribuídos e de vídeo vigilância a estações meteorológicas, monitorando fluxos de tráfego e fornecendo informações em tempo real ou em antecipação a riscos. Ou seja a segurança ubíqua se estende desde pequenos projetos, por exemplo, complexos comerciais ou de negócios com sistemas integrados de controle de edifícios, combinando vigilância por vídeo, detecção de incêndio e monitoramento de fluxo de multidões ou outros tipos de rastreamento de clientes, por meio de iniciativas maiores, mas temporárias, como os sistemas de comando e controle estabelecidos para esportes.

Uma cidade inteligente é desenvolvida, implantada e mantida com a ajuda da IoT e está sujeita ao desenvolvimento central de arquiteturas de segurança. Essa definição fornece uma

visão concisa das ameaças, vulnerabilidades e segurança disponíveis discutindo as melhores práticas existentes sobre segurança de cidades inteligentes (IJAZ et al., 2016). Para HALL et al. (2000) A segurança pode ser projetada, construída, e mantida usando tecnologias integradas altamente avançadas, que incluem sensores, eletrônicos e redes que estão ligados a sistemas informatizados compostos por bancos de dados, algoritmos de rastreamento e tomada de decisão.

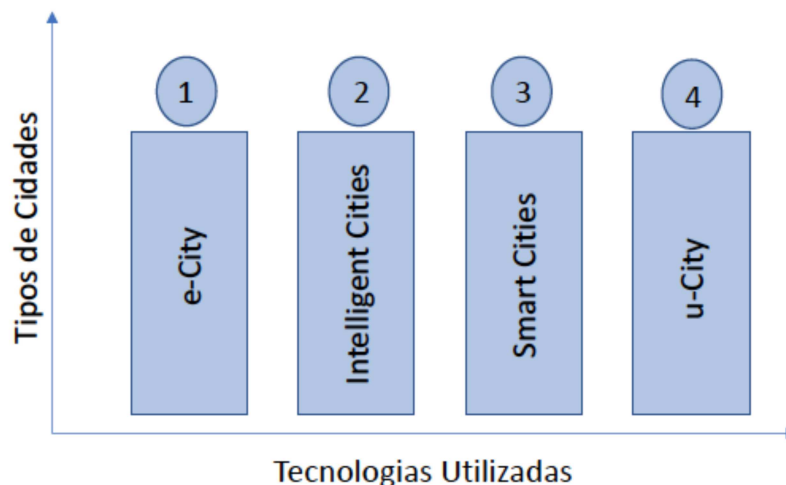
AL-MOTWAKEL e ZAHARY (2019) consideram que a computação ubíqua é cada vez mais reconhecida como uma séria preocupação mundial em segurança pública e privacidade. Atualmente, há cada vez mais invenções de dispositivos inteligentes as quais contribuem com a computação onipresente e tornarão possível a computação sempre e em qualquer lugar.

Fornecer segurança ubíqua implica em considerar uma área definida a fim de oferecer cobertura pelo sistema. Os trabalhos relacionados se dividem em diferentes formas de implementar isso. Inicialmente, surgem preocupações de segurança e privacidade, pois todo o controle da segurança ubíqua não apenas coleta uma grande variedade de informações sensíveis à privacidade das pessoas e de seus círculos sociais, mas também controla as instalações da cidade e influenciam a vida das pessoas.

2.4 Classificação das Cidades

Na Figura 9 são identificados os tipos de cidades e apresentada de forma simples uma classificação, tendo como base os tipos de tecnologia que cada modelo adota. Em uma *e-City*, ou *digital city*, tem-se o modelo mais básico, até a evolução dos modelos, chegando-se na *ubiquitous city*. O modelo identificado pelo círculo de número 3 representa uma cidade inteligente, onde neste nível encontram-se todas as tecnologias necessárias para a aplicação do modelo U-Follower.

Figura 9: Classificação das Cidades



Fonte: Elaborado pelo Autor.

No tipo de cidade apresentando na primeira classificação, tem-se o modelo *e-City* (COCCHIA, 2014; LEE; HANCOCK; HU, 2014) onde uma combinação de Internet de banda larga e infraestruturas são a base de conexão de toda uma comunidade. A utilização de serviços,

padrões e infraestrutura flexível atende um propósito geral, seja ao governo, cidadãos e empresas.

Na segunda classificação, tem-se as *Intelligent Cities* KOMNINOS (2002) para este tipo de classificação, as cidades são apresentadas como territórios dentro de um sistema inovador na mesma região. Este sistema faz a combinação entre seus indivíduos, instituições e espaços de inovação, estes, quase que na totalidade virtuais, para deixar mais fácil a gestão do conhecimento.

A terceira classificação traz o conceito de cidade inteligente ou cidade inteligente (PETROLO; LOSCRÍ; MITTON, 2014), este modelo tem por objetivo tornar os componentes de infraestrutura e serviços essenciais de uma cidade mais inteligentes, interligados e eficientes. Dentre os objetivos principais de uma cidade inteligente está o de melhorar a qualidade de vida, levando-se em consideração diferentes pontos de vista, tais como: o estado atual dos recursos, a consulta desses recursos, e o nível de acesso às informações. A quarta classificação é *u-City* ou *Ubiquitous City*, é resultado de minimização dos custos de banda larga, comercialização de sistemas de informações complexos, implantação de nuvem serviços e computação onipresente. As *u-City* estão sendo implementado na Coréia do Sul, Abu Dhabi (Cidade de Masdar) e Japão (Osaka) e entregam informações a qualquer momento, de qualquer lugar para qualquer pessoa, via Tecnologias da informação e comunicação (TIC) onipresente nas cidades (ANTHOPOULOS; FITSILIS, 2013).

2.5 Cidades Inteligentes

Conforme classificada na seção anterior, as iniciativas que compõem uma cidade *inteligente*, apresentam uma composição pertinente ao desenvolvimento de diversos serviços, entre eles, a segurança dos indivíduos dentro de um contexto urbano. As “*smart cities*” oferecem vários recursos disruptivos e possibilidades de investimentos (STRASSER; ALBAYRAK, 2016). Possibilitando o acesso detalhado a informações atualizadas, podendo-se comparar as decisões tomadas com aquelas que já se sabe anteriormente, possibilitando que se tome a melhor decisão. A segurança pública, dever do Estado, direito e responsabilidade de todos, é exercida para a preservação da ordem pública e da incolumidade das pessoas e do patrimônio.

A cidade de Porto Alegre é a oitava cidade mais inteligente do Brasil, segundo *Ranking Connected Smart Cities 2018* (CSC-19, 2018). E esta posição não foi alcançada à toa. O governo gaúcho lançou recentemente o *SmartGov*, um seminário de inovação com o objetivo de trazer para a pauta do Estado o processo de construção das *smart cities*. O entendimento é de que as soluções não são verdadeiras ou falsas, são boas ou ruins, e o desenvolvimento de uma cidade inteligente se refere ao processo de construção contínua da cidade (CSC-19, 2019). Logicamente, há muito ainda para ser construído.

Para uma cidade ser identificada como cidade inteligente, o investimento em sensores para o acesso e coleta de informações é necessário (PERBOLI et al., 2014). Estes sensores podem ser utilizados, por exemplo, para a coleta, em tempo real, de informações, possibilitando melhores decisões (BARBA et al., 2012; ALBINO; BERARDI; DANGELICO, 2015).

2.6 Considerações sobre o Capítulo

Este capítulo apresentou as tecnologias de computação em nuvem, computação ubíqua, IoT e *wearables*, todas fundamentais para a elaboração do modelo UFollower. O tema segurança ubíqua foi discutido, pois ele relaciona a segurança com as cidades inteligentes. Para apontar as tecnologias existentes nas cidades inteligentes, foi apresentada a classificação das cidades (COCCHIA, 2014; LEE; HANCOCK; HU, 2014). Dentre as classificações, foi destacada a cidade inteligente (PETROLO; LOSCRÍ; MITTON, 2014), pois seu nível de tecnologia cria um ambiente onde é possível a implantação da acessibilidade ubíqua, criando meios para surgimento do conceito cidade inteligente assistiva.

3 TRABALHOS RELACIONADOS

Este capítulo analisa trabalhos relacionados ao modelo proposto, levando em consideração pesquisas acadêmicas e projetos de instituições governamentais. Na análise foram incluídos trabalhos semelhantes ao modelo UFollower, mediante critérios que são especificados na seção 3.1. O restante do capítulo contém mais sete seções, da seção 3.2 até 3.6 são apresentados os trabalhos selecionados para análise. Na sétima seção 3.7 é realizada uma comparação entre os trabalhos. Por fim, a oitava seção, relata as considerações sobre o capítulo.

3.1 Critérios para escolha dos trabalhos

Os trabalhos foram selecionados com pesquisas nas bases de periódicos da CAPES, Wiley, *Scholar* – Google acadêmico, Elsevier – *Science Direct*, IEEE – *Xplore*, ACM – *Digital Library*, Springer – *International Publishing AG*, *dblp – computer science bibliography*, *Emerald Insight – Group Publishing Limited* e *CiteSeerx*. As pesquisas tiveram como base o termos de pesquisa e seus sinônimos, com a seguinte expressão: (“ubiquitous security” AND “assistive environments”) AND (“context aware” OR “location based services”) AND (“Smart Cities” AND “Surveillance System”) OR (“Smart Surveillance” AND “Smart City”).

A escolha dos trabalhos, levou em consideração se estes utilizavam computação ubíqua, ofereciam suporte voltado à segurança e se o trabalho considerava ambientes públicos dotados de algum tipo de tecnologia. Após as buscas nas bases citadas, os trabalhos foram submetidos a um comparativo, com base nos critérios: tipo de acessibilidade que oferece suporte, se utiliza perfil, trilhas, computação ubíqua (Computação em Nuvem, IoT e *Warables*), uso de dados de outros sistemas, oferta de dados para outros sistemas, monitoramento, ontologia ou algum componente para semântica, sistema de recomendações, ambiente dinâmico, busca de recursos e região de abrangência. Na seção 3.7 serão detalhados os critérios aqui mencionados. Nas próximas cinco seções serão apresentados os cinco trabalhos selecionados.

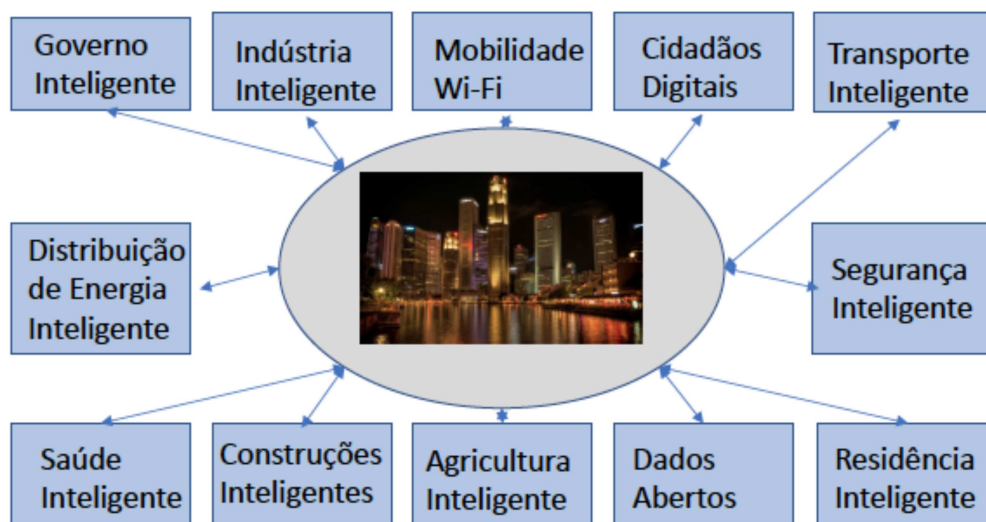
3.2 Privacy and Security Aspects of E-Government in Smart Cities

No trabalho de YANG, ELISA e ELIOT (2019) é proposto um modelo de IDS (do inglês, *Intrusion Detection System*) que integra um sistema de governo eletrônico a uma cidade inteligente para garantir a privacidade e segurança do usuário. O foco principal do trabalho foi avaliar a participação da esfera pública no formato de um sistema e-Gov nos aspectos de segurança e privacidade, tanto do indivíduo como das informações que por ele são geradas.

Para os autores, o governo eletrônico é parte indispensável em uma cidade inteligente que utiliza tecnologias de informação e comunicação com o objetivo de transformar as relações entre os órgãos do governo e os cidadãos, empresas e outros departamentos governamentais, visando melhores serviços públicos, interações aprimoradas e maior eficiência da operação governamental. O trabalho investiga a situação atual e as soluções técnicas disponíveis de sistemas de governo eletrônico em termos de segurança e preservação da privacidade no ambiente de cidades inteligentes. Ele também identifica os desafios na adoção de sistemas de governo eletrônico na implementação de cidades inteligentes.

Conforme pode ser verificado na Figura 10, o modelo levou em consideração diversos componentes de uma cidade inteligente em sua concepção.

Figura 10: Principais componentes de uma cidade inteligente



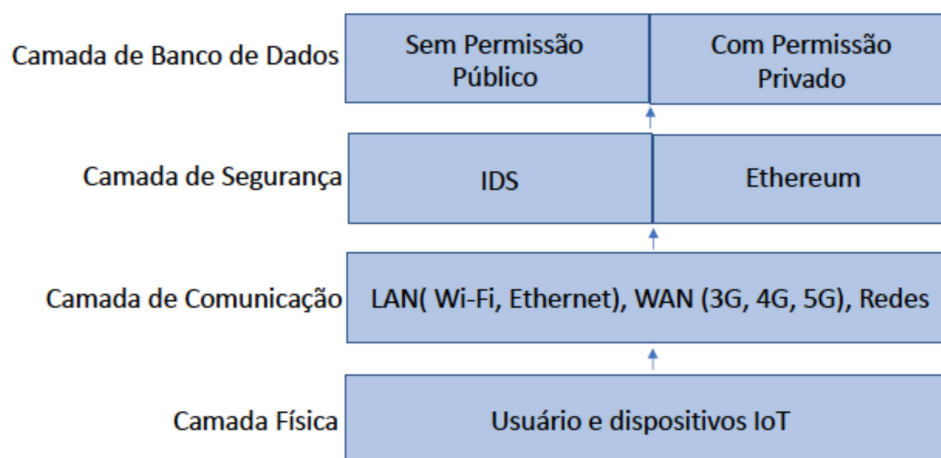
Fonte: Adaptado de YANG; ELISA; ELIOT (2019).

A Figura 10 trata de representar uma cidade inteligente cercada de diversas características e elementos que a tornam *smart*. Alguns desses itens são: mobilidade, e aqui a *Wi-Fi* é o destaque já fazendo a ligação com outro item que são os cidadãos digitais. A informatização é significativa para se atingir o objetivo de uma cidade inteligente. Seguindo nos itens, também é ilustrado o transporte inteligente, saúde inteligente, dados abertos e residência inteligente, aqui, com a utilização de IoT. Para concluir, dois elementos também representados na Figura 10 e que convergem diretamente para a proposta do modelo Ufollower: um governo inteligente juntamente com a segurança inteligente. Este dois últimos itens, subsidiados pela infraestrutura dos demais (YANG; ELISA; ELIOT, 2019).

Uma cidade inteligente geralmente acomoda vários órgãos de governança, composta por cidadãos, cidade, autoridades, empresas, indústrias e grupos comunitários e outros setores públicos e privados. A governança inteligente é um dos produtos do governo eletrônico que oferece participação eletrônica eficiente, serviços eletrônicos e administração pública, tornando-a um dos componentes de uma cidade inteligente. O governo eletrônico está mantendo o futuro das cidades inteligentes, pois todas as autoridades públicas locais que formam o governo em todo o mundo estão conectadas aos sistemas de governo eletrônico para preparar e permitir uma globalização mais profunda.

A validação do modelo foi simulada um ambiente de governo eletrônico utilizando técnicas de *blockchain* e IA na plataforma *Ethereum*. A seguir, a Figura 11 apresenta as camadas do modelo IDS.

Figura 11: Camadas do modelo IDS



Fonte: Adaptado de YANG; ELISA; ELIOT (2019).

Conforme pode ser verificado na Figura 11, a primeira camada é a camada física com dispositivos, cidadãos, empresas e funcionários, onde as funções básicas é a de captura de dados. A segunda camada é a de comunicação que se conecta entre os usuários e o sistema do governo, capturando e transmitindo os dados. A terceira camada é uma camada de segurança onde as medidas de segurança são implementadas na análise do tráfego para atividades suspeitas. A quarta camada abriga o banco de dado distribuído.

O trabalho concluiu que uma comunicação de informações *on-line* mais sensível, trazida pelo sistema de governo eletrônico, aumenta a segurança e privacidade associadas ao governo eletrônico e às cidades inteligentes. Com base no estudo realizado, a tecnologia pode garantir a privacidade, a segurança e a integridade de informações confidenciais. O modelo não foi posto em um ambiente de produção porque foi entendido que é necessária a avaliação sistemática do sistema proposto antes de sua implantação.

3.3 SPATH: Finding the Safest Walking Path in Smart Cities

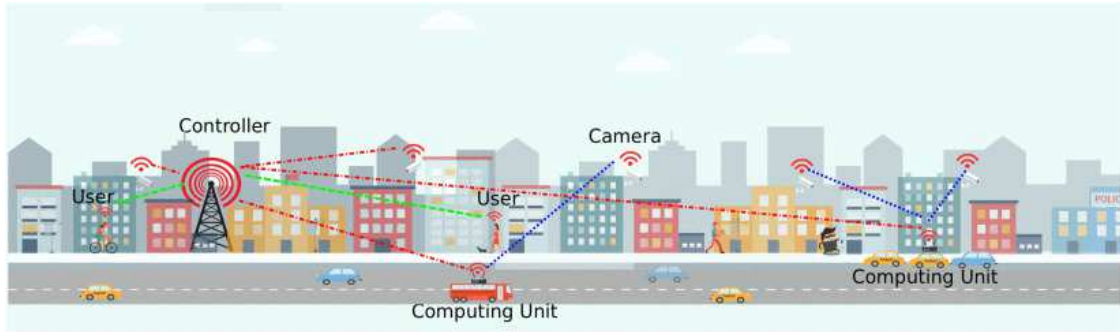
O trabalho de PANG et al. (2019) apresentou o aplicativo SPATH (*Safest PATH*, ou seja, o Caminha mais Seguro) para dar suporte de segurança ao cidadão americano, garantindo uma navegação segura, caminhando, em cidades inteligentes. Dado o fato de que mais de um milhão de crimes acontecem nos Estados Unidos da América todos os anos, a segurança pública se torna uma das preocupações mais importantes. Embora muitas aplicações relacionadas à segurança pública tenham sido desenvolvidas, garantir a segurança de caminhar até um destino, especialmente em uma cidade desconhecida, ainda é um desafio.

Para oferecer suporte a esse serviço, o SPATH utiliza-se de câmeras sem fio, infraestrutura de celular existente e veículos com recursos de computação subutilizados para processar e transmitir vídeos de vigilância, que podem ser visualizados pelos usuários para verificar o status atual de segurança do caminho dos pedestres.

A transmissão de longa distância de um grande volume de arquivos de vídeo pode causar congestionamento na rede. Para auxiliar nesta demanda, a tecnologia de resumo de vídeo, que é realizada com a utilização da capacidade de computação subutilizada em veículos, é aplicada

para extrair informações valiosas de um arquivo de vídeo enquanto compacta efetivamente seu tamanho. A Figura 12 ilustra como a comunicação na cidade inteligente é prevista pelo aplicativo SPATH. Nela, é possível verificar que os diversos nodos da cidade se comunicam através de *Wi-Fi*: O prédio está em rede com a câmera de outro prédio, que por sua vez está em rede com uma unidade computacional, na figura exemplificado pelo ônibus, todos estes elementos estando em rede com os usuário: pedestre, ciclista e passageiros.

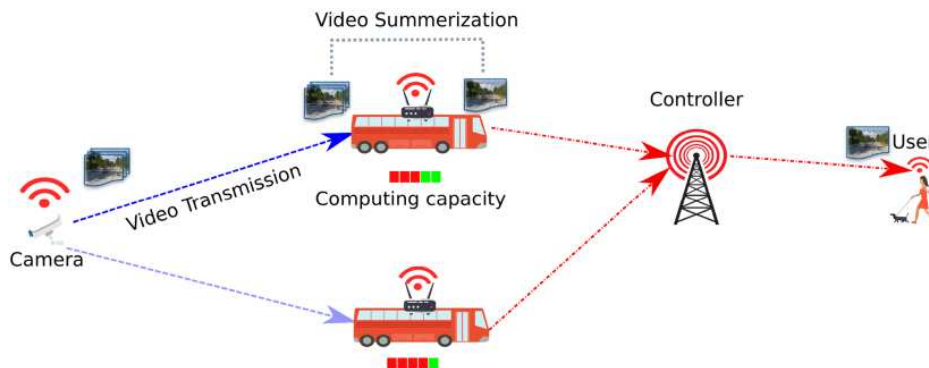
Figura 12: Arquitetura SPATH



Fonte: PANG et al. (2019).

A Figura 13 apresenta um usuário utilizando o SPATH, onde já selecionou o caminho que deseja percorrer caminhando. O aplicativo ativará as câmeras previstas pelo caminho (previamente cadastradas) para capturar vídeos de acordo com o caminho do usuário. Então, o aplicativo reúne informações relacionadas à comunicação, computação e segurança (índice de crimes) para tomar a decisão monitorando da maneira mais segura possível o percurso. De acordo com as informações de controle, câmeras pelo caminho transmitem os vídeos para as unidades locais de computação associadas. Por fim, os vídeos são entregues por meio da infraestrutura existente. Se o usuário considerar que o caminho a pé não é seguro o suficiente, ele poderá selecionar caminhos alternativos e fazer a solicitação do serviço novamente.

Figura 13: Ilustração de resumos e transmissões de vídeo



Fonte: PANG et al. (2019).

O trabalho utilizou as infraestruturas de celulares para transmitir vídeos de vigilância de câmeras de rua para os usuários com a intenção de identificar o *status* de segurança. Para lidar o grande volume de vídeos, aproveitaram os veículos com recursos de computação subutilizados resumir vídeos. Além disso, desenvolveram um conjunto de tarefas dispostas em um algoritmo de correspondência iterativa rápida (em inglês: *Fast Iterative Matching* - FIM) para resolver a otimização da latência. Os resultados da simulação mostram que a proposta atendeu ao que foi inicialmente projetado.

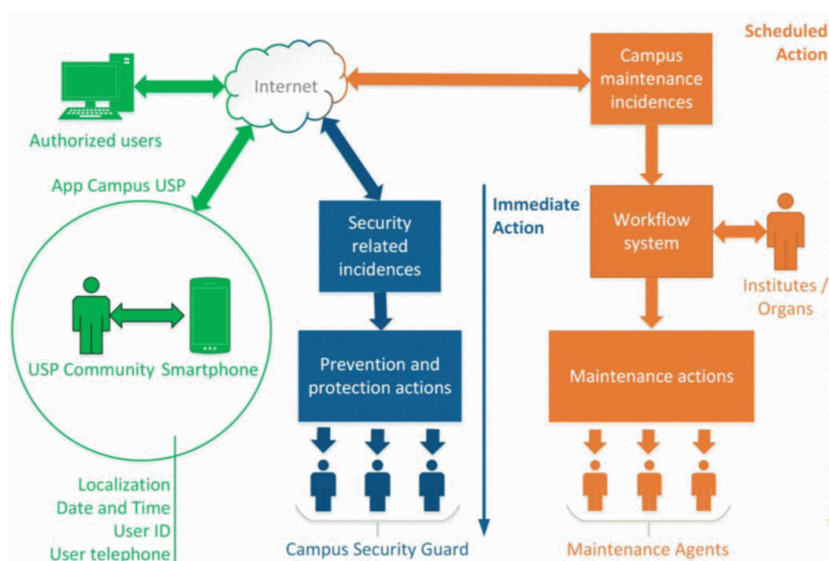
3.4 Smart services: A case Study on smarter public safety by a mobile app for University of São Paulo

A Universidade de São Paulo é a maior universidade do Brasil, distribuída por onze campi, incluindo quatro museus, dois hospitais, com uma área total superior a 76 milhões m², tendo quase 90.000 alunos de graduação e pós-graduação, cerca de 6.000 professores e 14.000 funcionários técnico-administrativos. Não à toa é chamada de cidade universitária. A equipe de segurança possui um centro de operação 24 por 7. A rotina inclui rondas preventivas no terreno do campus, resposta a acidentes de trânsito dentro do campus e resposta para a comunidade da USP em caso de emergências. Devido as dimensões de alguns dos campi a resposta imediata as emergências dos usuários nem sempre é possível.

O modelo de FERREIRA et al. (2018) propõem o Campus USP, uma solução *mobile* para a Universidade de São Paulo, em virtude dela apresentar problemas de segurança pública ao longo dos anos. A solução envolve um ambiente de segurança pública com detalhes da implementação e apresenta os dados estatísticos coletados pelo sistema, mostrando sua eficácia e conclui mostrando as melhorias introduzidas na segurança e bem-estar da comunidade universitária.

Em um esforço para criar um sistema inteligente de segurança pública em todos campi da universidade, um sistema inteligente de câmeras de vigilância foi desenvolvido e implantado. O aplicativo Campus USP apresenta alguns benefícios na interação com as equipes de resposta, sendo: a localização atual do usuário por GPS (do inglês, *Global Positioning System*), a hora atual do relatório, o envio de informações visuais usando a câmera, o envio de informações de áudio e texto e o contato diretamente com o centro de despacho de segurança por telefone.

Figura 14: Estrutura do aplicativo Campus USP



Fonte: FERREIRA et al. (2018).

A Figura 14 representa os componentes da estrutura dos sistemas do campus e como o aplicativo Campus USP está inserido nele. Conforme pode ser verificado, a comunidade acadêmica, representada por estudantes, professores e técnicos -funcionários administrativos, usa seu próprio dispositivo para se conectar através da Internet. O aplicativo móvel foi implementado nas plataformas iOS e Android. Quando o aplicativo móvel é usado pela primeira vez um texto explicando como o aplicativo funciona e quais informações do usuário estão sendo enviadas são apresentadas ao efetuar *login* no sistema. A interface principal possui apenas 4 botões: Relatar Incidência, Ligação de emergência, Mapa de segurança e “Vigie-me”.

O aplicativo móvel Campus USP foi utilizado por quase um ano, com cerca de 12.000 instalações nos sistemas iOS e Android. O tempo de atendimento de uma ocorrência pela equipe que realiza a ronda no campus era de cerca de 10 minutos antes da implantação do aplicativo. Após a introdução do aplicativo, foi reduzido para 5 a 7 minutos. Essa redução foi possível porque quando o usuário utiliza o aplicativo, sua localização e identificação são exibidas no centro de expedição para que o operador pode pular algumas perguntas e despachar o mais próximo veículo para o local do incidente quase imediatamente.

3.5 Urban Infrastructure safety system based on mobile crowdsensing

No trabalho de ZHAO et al. (2018) utiliza o conceito de *crowdsensing*, que consiste em uma técnica em que um grande grupo de indivíduos com dispositivos móveis compartilham coletivamente dados e extraem informações para medir, mapear, analisar, estimar ou inferir qualquer processo de interesse comum. O *crowdsensing* facilitou aplicações ubíquas de detecção móvel entre humanos e o mundo físico circundante como uma tecnologia de detecção conveniente e econômica. O estudo apresentou um sistema de monitoramento de segurança de infraestrutura urbana, conhecido como *Urban Safety*, que possibilita a coleta de informações sobre danos à infraestrutura urbana para agentes públicos e permite o monitoramento de emergência e avaliação no campo da prevenção e mitigação de desastres.

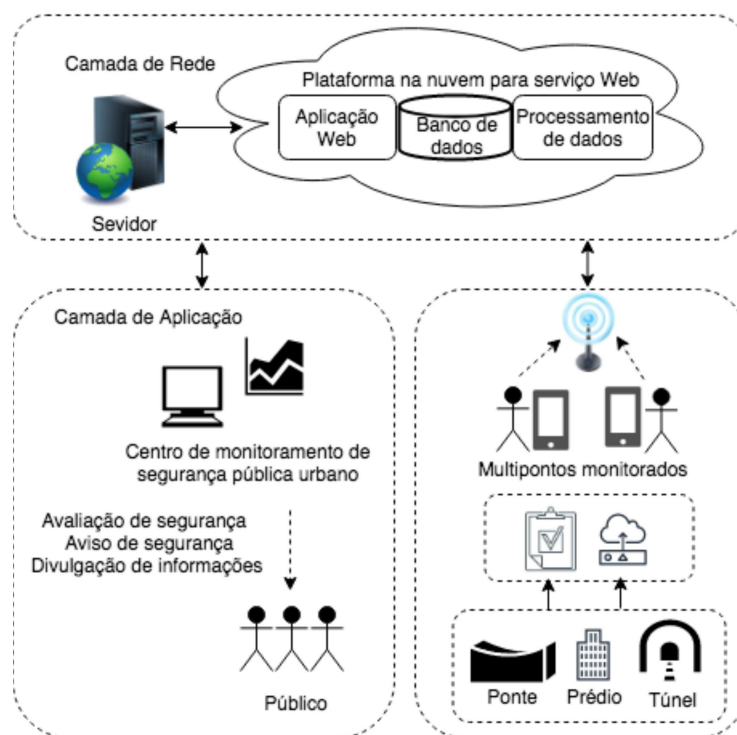
Como o entendimento de segurança vai além da segurança do indivíduo ou de seu patrimônio, o aplicativo proposto visa contribuir com a participação de todos os usuários na segurança de edificações e construções.

O aplicativo *Urban Safety* foi desenvolvido e apresentado com base na plataforma *Android*. O aplicativo atua como um sensor coletor de dados urbanos, enviando as coletas para um banco de dados de segurança urbana que armazena os dados detectados e enviados pelos usuários. Experimentos de verificação e avaliação foram realizados no Universidade de Tecnologia de Dalian (do inglês, *Dalian University of Technology-DLUT*), incluindo monitoramento de deslocamento, medições de dados e de pontos e coletas. Finalmente, os resultados experimentais mostram a uso viável e eficaz do aplicativo para o monitoramento de informações de segurança.

Atualmente, a quantidade de *smartphones* globais atingiu quatro bilhões. Portanto, a onipresença dos *smartphones* facilita a participação do público na detecção de incidentes. O *mobile crowdsensing* (MCS) é um novo paradigma para fazer com que *smartphones* possam detectar, coletar, fazer *upload* e analisar os dados em uma grande região de cobertura. Portanto, com base na tecnologia MCS, os *smartphones* podem ser amplamente utilizados em coleta de *big data* relevante para a segurança e infraestrutura urbana.

A Figura 15 apresenta como foi proposta a arquitetura do *Urban Safety*.

Figura 15: Arquitetura do sistema *Urban Safety*



Fonte: adaptado de ZHAO et al. (2018).

Conforme pode ser verificado na Figura 15, a arquitetura do aplicativo do *Urban Safety* é dividida em três camadas. Na camada de rede, está a nuvem onde os dados da aplicação são processados e incluídos no banco de dados. Esta camada também comporta a aplicação *Web*.

Na camada de aplicação, também há o processamento de dados e gerenciamento. Na camada de detecção, ocorre o monitoramento multiponto, com os usuários utilizando, de fato, o aplicativo.

A principal função da camada de detecção é usar o *smartphone* para monitorar as informações de risco de segurança das infraestruturas urbanas. Na camada de detecção, cada telefone é considerado um nó do sensor, que pode chamar os sensores internos, como acelerômetros, giroscópios e sensores de proximidade. Baseado em *smartphones*, a multidão pode perceber o monitoramento de coordenação de ponto único e multiponto para pontes, estradas, encostas, túneis e outras infraestruturas. A aceleração, ângulo, deslocamento, rachadura e outras informações de risco de segurança.

A camada de rede é a plataforma em nuvem e fornece uma plataforma de coordenação central para armazenar e integrar os diversos dados obtidos dos usuários. Além disso, a camada de rede fornece três serviços em nuvem para pessoas diferentes: infraestruturas como serviço (IaaS), plataforma como serviço (PaaS) e *software* como serviço (SaaS). Durante a análise de dados, alguns especialistas foram convidados a oferecer suas avaliações profissionais para garantir que os resultados sejam autênticos e racionais. Por fim, os dados processados foram carregados no servidor e o informações de risco de segurança puderam ser enviadas para a camada do aplicativo. O último componente do *Urban Safety* é a camada de aplicação, que é o centro de monitoramento de segurança para gerenciar a operação geral da rede, processar os dados e publicar as informações ao público. A tarefa de controle fornece o *design* de tarefas, designações de tarefas e sistema de gestão. O módulo de processamento de dados oferece avaliação e o aviso de segurança para informações sobre riscos à segurança obtidas na camada de rede. Além disso, o centro de liberação de informações fornece *feedback* imediato ao público.

3.6 Smart surveillance em aplicações recentes no Brasil: um estudo de caso nas cidades de Recife e Curitiba

O trabalho de BATISTA et al. (2016) trouxe o conceito de *smart surveillance*, contextualizado a partir da noção de smart cities. A prática de coletar informações do meio evoluiu ao longo do tempo, e o advento tecnológico dos últimos anos gerou mudanças substanciais neste processo. A utilização de tecnologias cada vez menores e mais sofisticadas para fins de vigilância, prevenção e militarização do espaço urbano é parte deste processo, e constitui a chamada *smart surveillance*. A pesquisa objetivou identificar de que modo o aparato tecnológico para vigilância instalado vem sendo utilizado nas cidades de Recife e Curitiba, no sentido de aproveitar o legado do eventos esportivos, como a Copa do Mundo de 2014, por exemplo, e de contribuir para a gestão tecnológica do espaço urbano e para a redução da criminalidade.

Os resultados indicaram que em ambas as cidades, a implantação dos Centro Integrados de Comando e Controle se assemelha a um tipo moderno de panóptico, o qual está baseado no discurso do combate aos problemas sociais e à violência urbana, além da relativa isenção da liberdade e do anonimato. Os conceitos de segurança, defesa e vigilância, bem como a prática de coletar informações do meio passaram por evoluções ao longo do tempo. O advento tecnológico dos últimos anos gerou mudanças substanciais, como uma maior variedade das informações disponíveis, maior acessibilidade de coleta e a duração perpétua do registro.

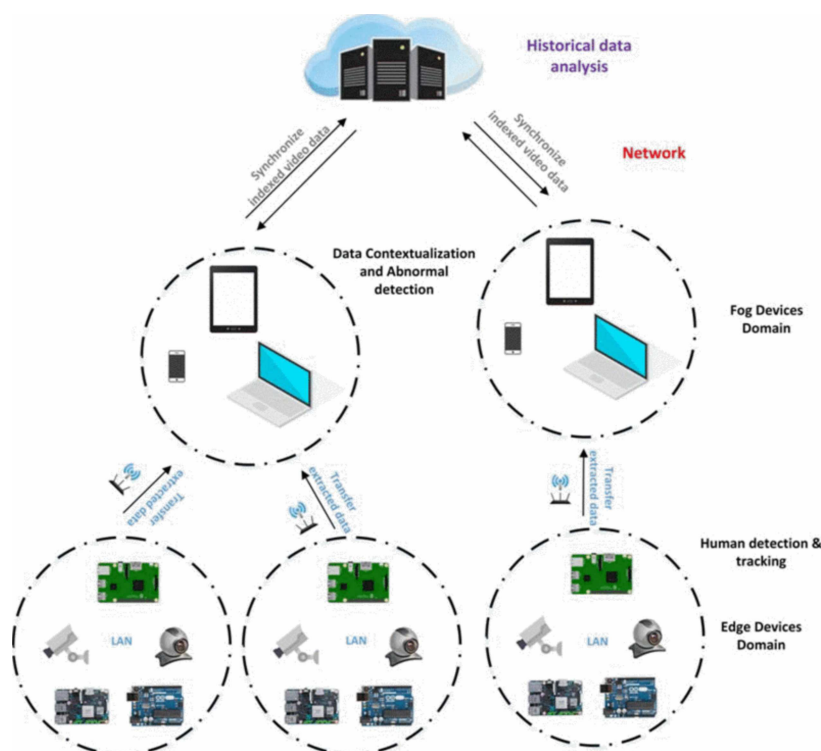
A utilização de tecnologias – cada vez menores e mais sofisticadas – para fins de vigilância, prevenção e militarização do espaço urbano é parte da construção dos chamados

smart spaces e/ou *smart cities* (espaços inteligentes e cidades inteligentes, em tradução livre, respectivamente, embora a palavra “inteligente” não reflita, em totalidade, a complexidade presente no termo “*smart*”). As *smart cities* são, de acordo com (KITCHIN, 2014), redes urbanas – complexas e de alta densidade – que podem ser gerenciadas a partir da utilização de tecnologias. Não são, portanto “cidades do futuro”, mas espaços que apresentam diversos desdobramentos sociais e espaciais a partir das conexões criadas por instrumentos digitais.

O processo de vigilância também pode passar pelo processo de se “tornar inteligente”, sendo então denominado *smart surveillance* (HAMPAPUR et al., 2004). De acordo com Bruno (2013b), a *smart surveillance* utiliza-se de tecnologias inteligentes que agem controlando diretamente o meio. Por utilizar câmeras “*smart*”, este processo pode ser menos invasivo, uma vez que é orientado para identificar mudanças em padrões de comportamento, não necessariamente precisando identificar cada indivíduo. Não é, no entanto, um processo menos perigoso do que as formas tradicionais de vigilância apenas por possuir essa característica.

O espaço cada vez mais tecnológico, repleto de aparatos cada vez mais sofisticados, invisíveis e ubíquos, altera a relação estabelecida entre os seres humanos e o meio, fazendo com que todos os elementos que compõem o cotidiano urbano sejam potencialmente capazes de tornarem-se dados em uma rede. A Figura 16 ilustra a arquitetura do *smart surveillance*.

Figura 16: Arquitetura *smart surveillance*



Fonte: CHEN (2000).

A noção de privacidade baseada na dicotomia do público e do privado vem sendo alterada. Neste contexto, a imagem atual dos processos de vigilância é de que se trata de algo onipresente e onipotente, assim como apresentado por Orwell em “1984” e por FOCAULT (1999), na teoria da sociedade disciplinar criada a partir do conceito de panóptico. Tal conceito foi desenvolvido por Bentham, a partir de um projeto arquitetônico desenhado como a prisão “ideal”. Trata-se de um edifício em forma de anel, onde se encontram as celas, com um pátio

central, onde fica a torre vigilante. Nessa construção, cada cela teria uma janela destinada apenas à entrada de ar e luz, a qual impediria, no entanto, que o prisioneiro pudesse ver o que está fora. (GUNDALINI, 2013). O projeto induz à ideia de constante vigilância e inspeção, a partir do isolamento dos detentos. Desse modo, o conceito de panóptico estabelece um poder de vigilância mais potente do que o olhar humano. Não se trata, portanto, apenas de uma ideia de arquitetura como controle social; influencia, também, o organizacional e espacialmente o modo de agir dos observados e dos observadores.

Assim, emerge o conceito de *smart surveillance*, em que os sistemas de monitoramento seguem, em tempo real, algoritmos determinados para automatizar a identificação e o tratamento de ocorrências (detectando as situações, reconhecendo e analisando comportamentos) (DURÃES, 2008).

3.7 Análise comparativa e contribuição do modelo UFollower

Os trabalhos apresentados nas seções anteriores foram selecionados para análise e são considerados trabalhos relacionados ao modelo UFollower, pois oferecem recursos de segurança e exploram outros conceitos usados neste trabalho. Para realizar a análise comparativa entre os trabalhos foram estabelecidos critérios que são apresentados a seguir.

- Aplicado a Cidades Inteligentes: identifica se o trabalho apresentado é sobre ou pode ser aplicado em cidades inteligentes. Nesse critério os trabalhos foram classificados como: “*survey*”, “já implementado” ou “restrito”. A classificação “*survey*” foi atribuída para os trabalhos que fizeram uma pesquisa atual sobre segurança aplicada em cidade inteligente.
- Monitoramento: Identifica se o trabalho monitora constantemente o usuário e oferece uma *smart surveillance* para melhor atendê-lo, principalmente, no quesito segurança. A monitoração consiste em realizar uma análise de seu contexto, das ações feitas, dos lugares acessados e dos objetos ao seu entorno, a fim de identificar possíveis situações de desconforto, perigo ou risco. Caso o monitoramento detectar algo que deve ser informado, isto é realizado. Este critério recebeu a classificação “sim”, para os trabalhos que realizam tal operação, “parcial” para os trabalhos que monitoram, mas não oferecem as informações instantaneamente, ou “não” para os trabalhos que não realizam monitoramento.
- Computação Ubíqua: identifica se o trabalho adota as tecnologias consideradas no capítulo 2, ou seja, 1) computação em nuvem, 2) *Wearables*, 3) IoT. E 4) Histórico de contexto.
- Aspectos de Segurança: Conforme já descrito o UFollower trata de ser, entre outras características, um agente ubíquo de segurança. Portanto, este requisito identifica se o trabalho contribui de alguma forma para a segurança do indivíduo, seja ela pública ou privada.
- Compartilhamento: identifica se, utilizando de computação ubíqua (computação móvel) os usuários interagem entre si compartilhando coletivamente dados e extraíndo informações para medir, mapear, analisar, estimar ou inferir qualquer processo de interesse comum. Se é um aplicativo, sim, senão não.
- Recomendações: identifica se o modelo analisa as escolhas feitas pelos usuários, identificando padrões de comportamento, a fim de auxiliar o usuário observado ou demais usuários nas situações futuras, fazendo recomendações, utilizando

mineração da dados e *big data*. As classificações atribuídas para este critério foram “adota” e “não adota”.

- Computação em Nuvem: conforme descrito na seção 2.1.1, trata de identificar se o trabalho proposto utiliza-se de computação em nuvem para o compartilhamento das informações.
- Ambiente dinâmico: identifica se algum componente do modelo coleta informações sobre o ambiente DEY (2001) e disponibiliza o estado atual do mesmo. Neste critério também foi considerado se o modelo contempla rede de sensores e IoT. As classificações atribuídas para este critério foram “estático”, “dinâmico”, “ausente”, sendo que a classificação “estático” foi atribuída para modelos que disponibilizam informações sobre o ambiente, mas não oferece detalhamento oferecido por sensores *in loco*. Já a classificação “dinâmico” foi atribuída aos modelos que capturam informações. A classificação “ausente”, foi atribuída aos modelos que não compartilham informações coletadas por sensores.
- Histórico de contexto: identifica se o trabalho gerencia e utiliza histórico das ações realizadas para e pelos usuários. Neste critério os trabalhos receberam a classificação “utiliza”, “parcial” ou “não utiliza”.

A Tabela 1 apresenta a comparação entre os trabalhos relacionados.

Tabela 1: Comparativo entre os trabalhos relacionados

Critério	<i>IDS</i> Modelo	<i>SPATH</i>	<i>Campus</i> <i>USP</i>	<i>Urban</i> <i>Safety</i>	<i>Smart</i> <i>Surveillance</i> - <i>CICC</i>
<i>1- Aplicado a Cidades Inteligentes</i>	extensível	completa	Restrito	completa	Parcial
<i>2-Monitoramento</i>	Não	Sim	Parcial	Parcial	Parcial
<i>3-Computação Ubíqua</i>	Não Utiliza	IoT	Não Utiliza	Computação em nuvem e IoT	Não utiliza
<i>4-Aspectos de Segurança</i>	Sim	Sim	Sim	Sim	Sim
<i>5-Compartilhamento</i>	Não	Sim	Sim	Sim	Não
<i>6-Recomendações</i>	Não	Adota	Adota	Adota	Não
<i>7-Computação em Nuvem</i>	Não	Não	Adota	Não	Não
<i>8-Ambiente Dinâmico</i>	Estático	Dinâmico	Dinâmico	Dinâmico	ausente
<i>9-Histórico de Contexto</i>	Não Utiliza	Não Utiliza	Não Utiliza	Não Utiliza	Não Utiliza

Fonte: Elaborado pelo Autor.

Com relação ao critério segurança, todos os trabalhos oferecem suporte alguns de maneira mais abrangente do que outros. Especificamente segurança pública, o trabalho *Campus USB* é mais restritivo do que o trabalho *Smart Surveillance-CICC* devido a sua área de abrangência ser mais reduzida. Com relação ao critério computação ubíqua, todos utilizam funcionalidades de serviços *Web* ou *mobile*, e apenas o trabalho *Urban Safety* utiliza serviços de computação em nuvem para integrar os dados obtidos dos usuários. Já as funcionalidades da IoT é utilizada pelo *SPATH* e *Urban Safety*. O recurso de *wearables* não é oferecido por

nenhum dos trabalhos. No critério monitoramento apenas o modelo ID não realiza tal operação, os trabalhos Campus USP, *Urban Safety* e *Smart Surveillance-CICC* fazem monitoramento parcial, ou seja, monitoram, mas não oferecem as informações instantaneamente (em tempo real). Já o trabalho SPATH monitora de forma completa. Com relação ao critério recomendações, apenas os trabalhos IDS Modelo e *Smart Surveillance-CICC* não contemplam este recuso. Com relação ao critério ambiente dinâmico apenas o trabalho *smart Surveillance-CICC* não adota sensores para gerenciamento do estado dos recursos, os trabalhos SPATH, Campus USP e *Urban Safety* oferecem informações dinâmicas dos recursos, ambientes e usuários. O trabalho DS Modelo, aborda este tipo de suporte no formato estático. Com relação ao critério Aplicado a Cidades Inteligentes, os trabalhos SPATH e *Urban Safety* abordam de forma completa. O trabalho IDS Modelo é extensível, o trabalho *Smart Surveillance-CICC* aborda de forma parcial e o único trabalho que aborda de forma restrita é o trabalho Campus USP, por se tratar apenas de perímetro da universidade. Com relação ao critério compartilhamento, os trabalhos que não adotam são IDS Modelo e *Smart Surveillance-CICC*.

Por fim, o suporte à computação em nuvem apenas o trabalho Campus USP adota, considerando o descrito na seção 2.1.1. Tendo em vista estas questões, é proposto um modelo para suporte à segurança, descrito no próximo capítulo.

A contribuição do modelo UFollower consiste na infraestrutura proporcionada, a qual pode ser aplicada em larga escala, atendendo sistemas com grande número de usuários nos diferentes tipos de possibilidade de segurança. A representação dos ambientes, usuários e recursos é constituída de forma semântica em função de utilizar ontologia. O modelo faz uso ainda de inferências bayesianas e histórico de contextos. O UFollowerServer contém as informações registradas pelo UFollowerNode, que é responsável pelo gerenciamento dos sensores instalados nos ambientes monitorados. Para gerenciamento dos usuários é adotado o conceito de perfis os quais serão implementados em um sistema multiagente (ARBSVU), que entre outras funções, deverá atuar junto ao agente de recomendação e o agente de recursos e locais. Os recursos são mantidos no modelo com informações de seu estado atual, por meio de rede de sensores (UFollowerNode). Os ambientes serão considerados para geração do contexto, e terão representação semântica também por meio da rede de sensores projetada.

Os recursos tais como tornozeleiras e sensores acústico de disparo, não são mapeados em nenhum dos trabalhos apresentados. No modelo UFollower, o recurso tornozeleira será mapeado, a fim de informar que tal recurso está à disposição. Entre os cenários simulados o uso das tornozeleiras foi utilizados no modelo As respostas para os critérios apresentados na seção 3.1, são respondidas pelo modelo UFollower. Tais respostas são apresentadas na Tabela 2.

Tabela 2: Respostas as critérios no modelo UFollower

<i>Critério</i>	<i>Proposta do Modelo UFollower</i>
<i>1-Aplicado a Cidades Inteligentes</i>	O modelo se aplica a regiões metropolitanas
<i>2-Monitoramento</i>	Agente ARBSVU realiza Monitoramento
<i>3-Computação Ubíqua</i>	Com tecnologias de Computação em Nuvem, IoT e <i>Wearables</i>
<i>4-Aspectos de Segurança</i>	Notificações tanto para o usuário quanto para a autoridade
<i>5-Compartilhamento</i>	Utilização coletiva dos usuários para efetivar segurança
<i>6-Recomendações</i>	Agente ARBSVU faz recomendações
<i>7-Computação em Nuvem</i>	Utiliza recursos públicos e privados
<i>8-Ambiente Dinâmico</i>	As informações oferecidas pelo UFollowernode
<i>9-Histórico de Contexto</i>	As informações oferecidas pelo meio (contexto)

Fonte: Elaborado pelo Autor.

3.8 Considerações sobre o capítulo

Neste capítulo foi realizado um estudo comparativo dos trabalhos relacionados com o modelo UFollower. Inicialmente foram realizadas buscas em bases de dados de periódicos nacionais e internacionais à procura de trabalhos que pudessem ser comparados com o modelo proposto. O objetivo do comparativo foi identificar as contribuições dos trabalhos propostos. Com base nesse comparativo foi possível conhecer oportunidades a serem exploradas na segurança ubíqua. A especificação de um modelo para segurança ubíqua exige a utilização e exploração de sensores e infraestrutura das cidades além do envolvimento de simulação para a validação do modelo. Mesmo diante de experimentos específicos apresentados pelos próprios autores, foi possível estabelecer um comparativo apresentado na Tabela 2, com base neste comparativo e nos estudos efetuados, o próximo capítulo propõe o modelo UFollower, com sua arquitetura e principais componentes.

4 MODELO UFOLLOWER

Este capítulo descreve o modelo para cidade inteligente voltado à segurança, denominado UFollower. O UFollower, entre outros, visa efetuar a coleta de informações de segurança do indivíduo de forma a otimizar a geração de históricos de contexto, obtendo informações do indivíduo, auxiliando-o em situações de risco. Este trabalho propõe desenvolver os agentes, o *backend*, *frontend* e a integração com um banco de dados. A primeira seção apresenta uma visão geral sobre o modelo e seus principais conceitos. A segunda seção apresenta requisitos da solução proposta. Na terceira seção é apresentada a arquitetura com os componentes do modelo, ontologia e agentes. A quarta seção apresentadas as considerações sobre o capítulo.

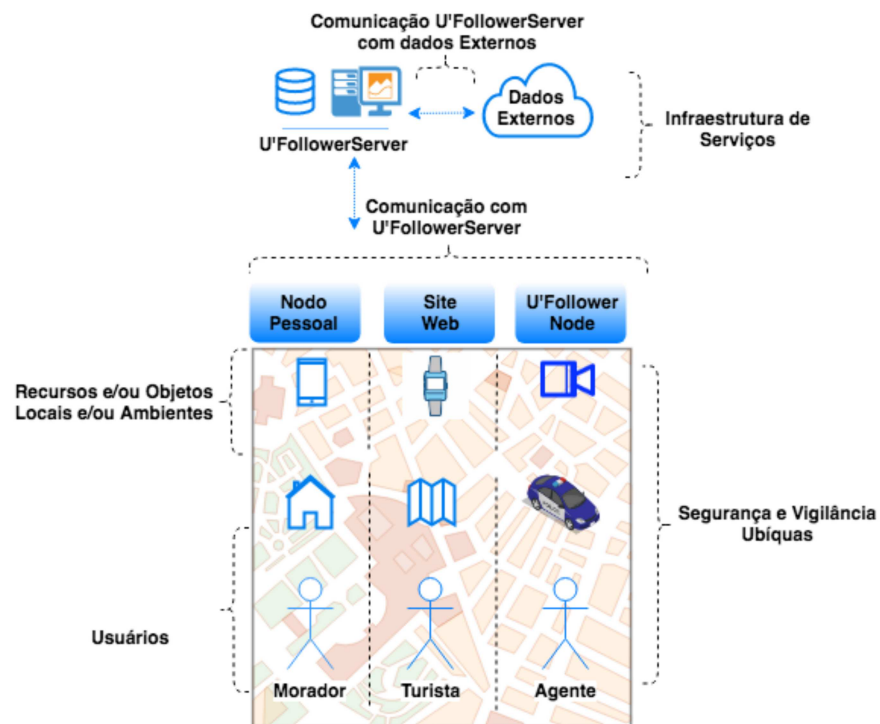
4.1 Visão geral do modelo

Nas cidades inteligentes, diversas tecnologias podem ser usadas para capacitar cidadãos e organizações no desenvolvimento de inovações e soluções colaborativas que tornam as cidades mais eficientes, sustentáveis e inclusivas. A realidade urbana da cidade inteligente emerge de redes de banda larga, aplicativos de *software*, dados de sensores e envolvimento do usuário, que tornam as cidades interativas, mensuráveis, inovadoras e que respondem em tempo real (KOMNINOS, 2018). Dentro do conceito de segurança e vigilância ubíquas, os fatores considerados pelo modelo UFollower, incluem a mobilidade urbana, a assistência de notificações e a oferta de informações sobre os recursos relevantes à segurança.

Para o funcionamento do UFollower são necessárias informações sobre os usuários, objetos e ambientes. Serão utilizados dados coletados pelo UFollowernode, *smartphones* e fontes externas, tais como: tornozeleiras, objetos vestíveis e câmeras. Na Figura 17 são apresentados os elementos que fazem parte do modelo UFollower, tais como usuários, locais, recursos, assistente pessoal, UFollowernode, UFollowerserver e dados externos. Os usuários são apresentados como atores na figura, podendo ser um usuário nativo da cidade (morador), um visitante (turista) ou um agente (policial, fiscal de trânsito, bombeiro, médico).

Quanto a região de abrangência, o modelo UFollower se destina a atender regiões metropolitanas. As regiões metropolitanas são adotadas, pois nelas é possível o uso de sensores fixos ou móveis para coletar informações necessárias ao modelo. Os sensores estão representados em um componente do modelo chamado de UFollowerNode, o qual realiza leitura de informações e envio para computação em nuvem. Mesmo que algumas regiões não sejam atendidas pela rede de sensores fixos, as mesmas podem ser vinculadas ao modelo, porém possuindo menos detalhes sobre o local, ou ainda possuindo informações apenas do momento em que um sensor móvel passou por este local. O UFollowerNode pode coletar informações de objetos identificados com *tags* e oferecer informações atualizadas para o agente de recomendações, dentro do UFollowerServer.

Figura 17: Visão geral do modelo UFollower



Fonte: Elaborado pelo Autor.

Na parte superior do mapa da Figura 17, são representados o assistente pessoal *NodoPessoal*, o *Site Web* e o dispositivo composto por sensores, *U'FollowerNode*. O *NodoPessoal* pode ser instalado em dispositivos móveis, podendo se comunicar com objetos, tais como um *wearable*, câmera ou sensor disponível no ambiente. O *U'FollowerNode*, é responsável pela coleta de informações contextuais em tempo real e envio destas ao *U'FollowerServer*. São propostos dois módulos de *U'FollowerNode*, um poderá ser fixado em algum ponto estratégico e será responsável pela coleta de informações contextuais sobre o local. O outro servirá como um módulo de retaguarda desse primeiro, onde a informações coletadas serão analisadas e repassadas a agentes de recomendação. Como o primeiro módulo necessita de instalação física para enviar informações relativas a posição atual do mesmo e não utiliza alimentação da rede elétrica, necessitará de reposição de baterias ao longo do tempo.

4.2 Requisitos do Modelo

Para atender as especificações do modelo UFollower são definidos os seguintes requisitos:

- Utilizar computação em nuvem, onde serão implementados servidores para os componentes do modelo;
- Utilizar uma ontologia para representação das entidades (usuários, objetos e ambientes);

- Monitorar as entidades, apresentar informações sobre a situação das mesmas e oferecer recomendações e sugestões em situações diversas;
- Permitir acesso ao modelo por meio de dispositivos móveis (*smartphone* e *tablet*), e demais computadores convencionais, tais como *notebook* e *desktop*;
- Integrar o modelo com recursos já utilizados na segurança, tais como câmeras e tornozeleiras eletrônicas;
- Permitir a integração de dados externos ao modelo, assim como oferecer dados para sistemas externos, por meio de chamadas *Web service*;
- Realizar a coleta de informações por meio de sensores via comunicação Wi-Fi;
- Utilizar IoT para monitoramento de objetos;
- Permitir a inserção de informações no sistema de forma automática e manual.

4.3 Arquitetura do modelo UFollower

A arquitetura do UFollower foi desenvolvida baseando-se na linguagem unificada de modelagem (UML do inglês *Unified Modeling Language*), fazendo-se uso de algumas definições do padrão TAM (*Technical Architecture Modeling*). Esta padronização é utilizada na SAP. Na Figura 17 é possível verificar a arquitetura geral do modelo. Nela constam os atores (Cliente Móvel, Cliente *Web*, Estado (UF-RS) e Dados Externos, blocos (Nodo Pessoal, *Web*, UFollowerNode e UFollowerServer).

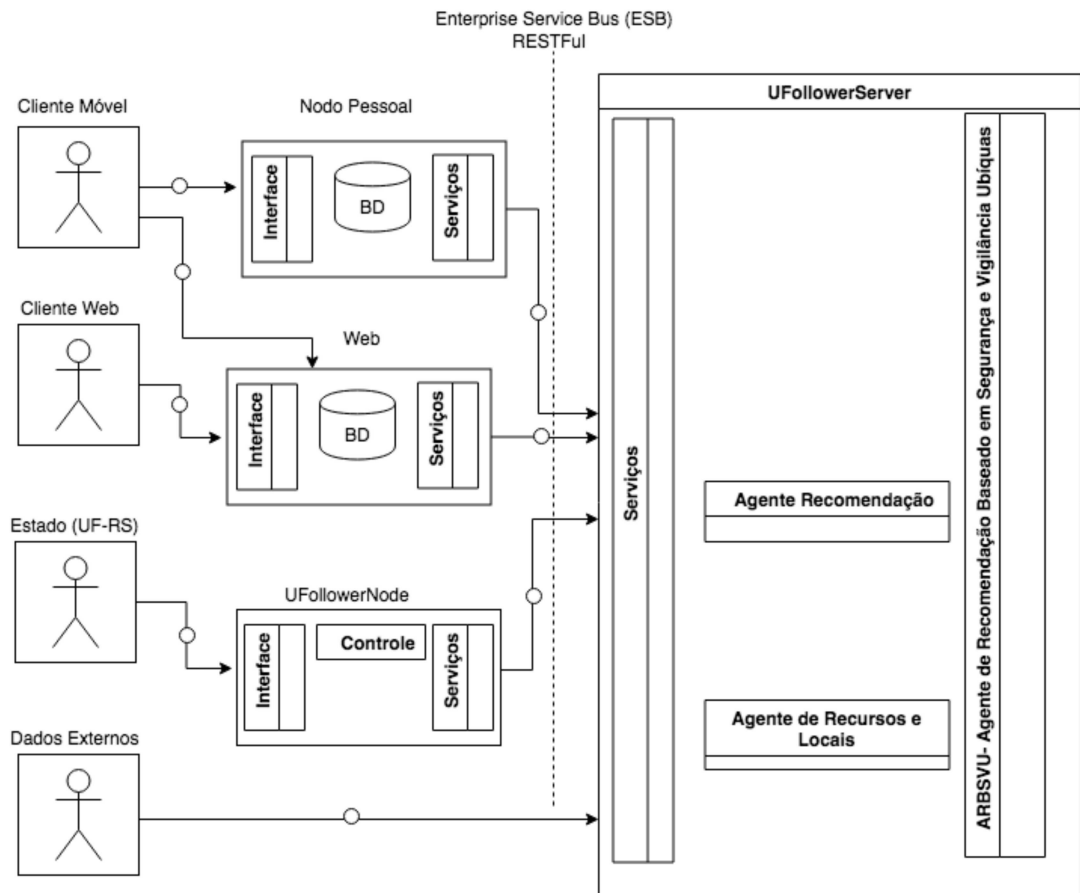
Além de se basear no modelo TAM, a arquitetura do modelo foi guiada pelo MASC, um modelo computacional para acessibilidade em cidades inteligentes), conforme projetado por (TELLES; BARBOSA; RIGHI, 2017).

O Nodo Pessoal refere-se ao assistente pessoal que um ator Cliente Móvel tenha acesso. Um exemplo pode ser um *smartphone*. Esta comunicação acontece fazendo uso de uma instância de *Web Service* (instWSNP), conforme pode ser verificado na Figura 18.

Os serviços nesta etapa estão disponíveis através da instância de *Web Service* instWSWeb. No Nodo *Web* é possível efetuar o cadastro, monitoramento e manutenção do UFollower. O Nodo UFollowerNode possui um *middleware* o qual tem a responsabilidade de atender o ator Estado (UF-RS).

O Nodo UFollowerServer comunica com todos os outros componentes. É o servidor da arquitetura. Toda a comunicação com o servidor é realizada por uma Arquitetura Orientada a Serviços (do inglês, *Service-oriented architecture*-SOA), através de um barramento de serviços (do inglês, *Enterprise Service Bus*- ESB). Este Nodo tem o seu próprio banco de dados para armazenamento de status e outras informações que precisam ser trocadas como clientes e tem o Agente de Recomendação Baseado em Segurança e Vigilância Ubíquas - ARBSVU. Este agente tem o papel de identificar o perfil do Nodo Pessoal que está interagindo com o UFollower, tornando a experiência de uso mais assertiva. A Figura 18 mostra a arquitetura do modelo.

Figura 18: Arquitetura do UFollower



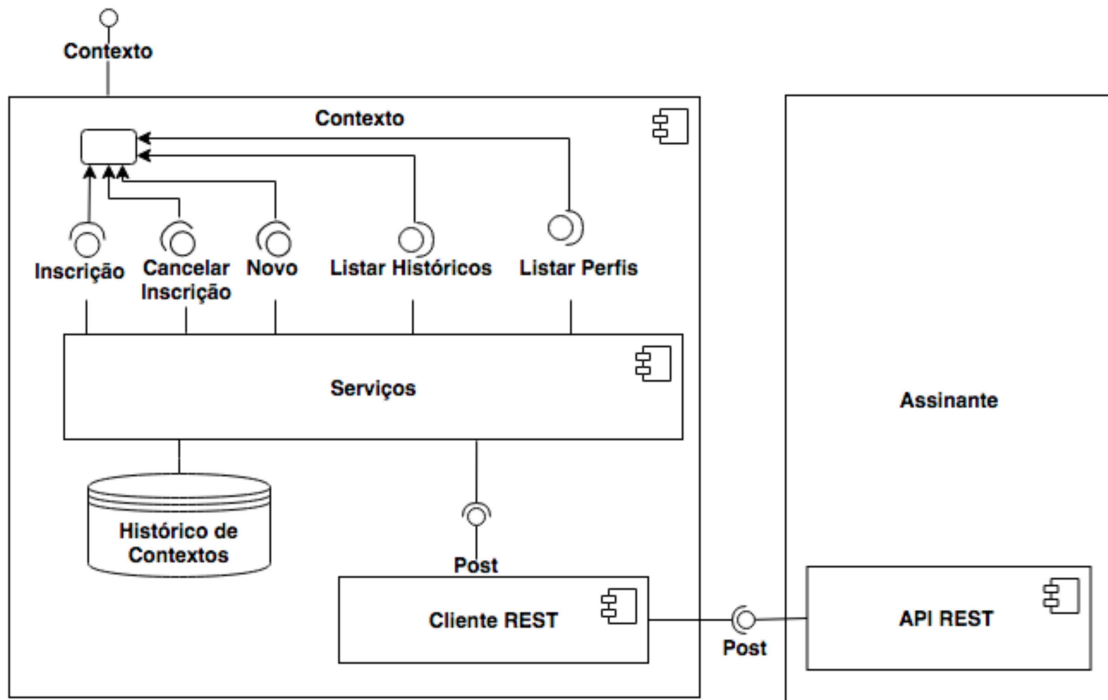
Fonte: Elaborado pelo Autor.

O componente de Contexto possibilita aos módulos do UFollower e clientes informarem seu interesse nos atributos que identificam o estado de um nodo pessoal em específico, e receberem notificações quando tais atributos forem alterados.

Desta forma, como exemplo, as informações de contexto que o modelo pode utilizar e guardar são: localização de um node pessoal, batimentos cardíacos de um usuário assinante utilizando um *wearable*, câmeras pública em vias e sensores.

Conforme pode ser verificado na Figura 19, o Componente de Contexto é parte fundamental para tratar os históricos de contextos e, juntamente com o Agente de Recomendação, utilizar a inferência de Redes Bayesianas para tratar as entradas e propor conclusões. Para os cenários que serão apresentados nesse trabalho, os contextos foram tratados tendo em vista a o vasto treinamento aplicados nos modelos e as entradas fornecidas pelos Nodos Pessoais dos usuários.

Figura 19: Componente de Contexto como parte do Agente de Recomendação

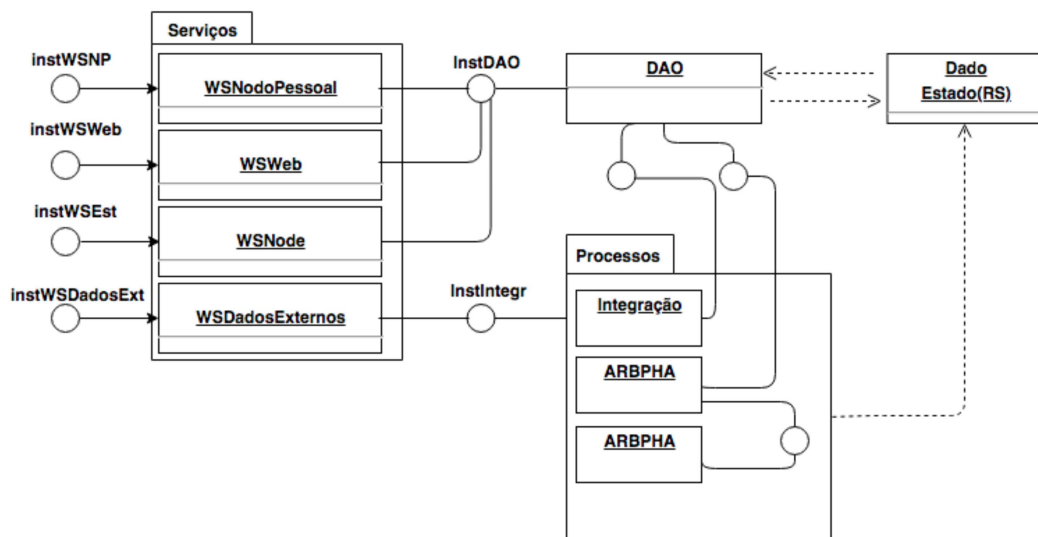


Fonte: Elaborado pelo Autor.

4.3.1 Servidor UFollowerServer

A Figura 20 apresenta o diagrama de componentes do servidor. No diagrama podem ser verificados os principais elementos que envolvem seu funcionamento.

Figura 20: Descrição do UFollowerServer



Fonte: Elaborado pelo Autor.

No diagrama de componentes do UFollower são expandidos os quatro blocos internos identificados como *Services*, DAO, Processos e Dado. Os *Web services* presentes no bloco Serviços acessam tanto o bloco DAO como o bloco Processos. As instâncias de *Web Services* instWSNP, instWSWeb, instWSEst e instWSDadosExt são utilizados para os respectivos acessos aos serviços: WSNodoPessoal, WSWeb, WSNodo e WSDadosExternos.

O componente DAO é responsável pelo acesso aos dados do modelo, tais dados estão armazenados em bancos de dados relacionais (RDBMS do inglês *Relational Database Management System*). O componente WSNodoPessoal atende às requisições feitas pelo Nodo Pessoal através da interface instWSNP. Dentre as requisições atendidas se destacam:

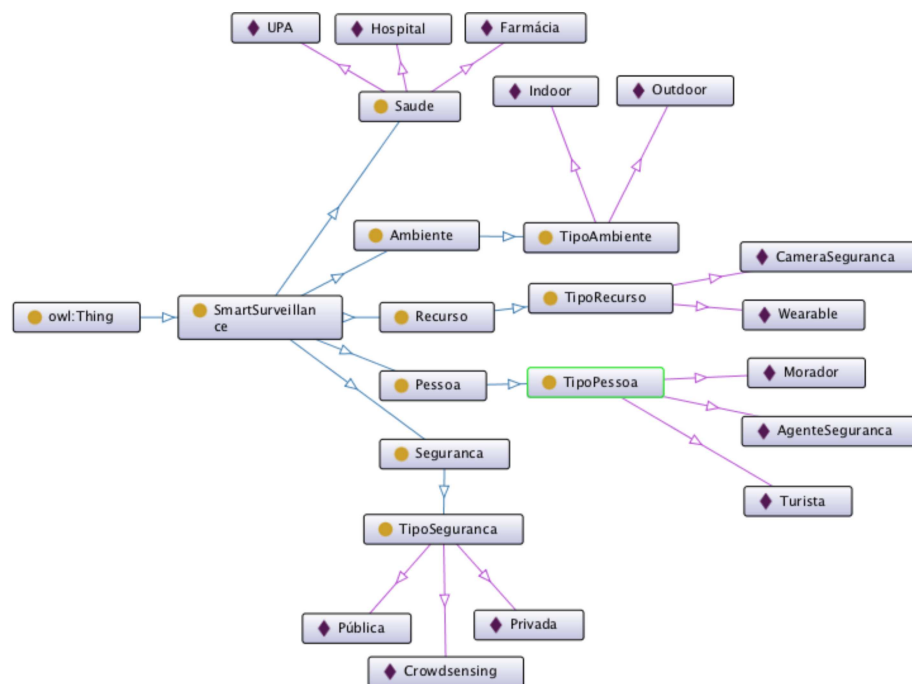
- Solicitação de *Login*: abre um formulário solicitando o usuário e a senha, uma vez informados é validado junto ao UFollowerServer e, se a autenticação for validada, o acesso é concedido. O componente tem ainda a possibilidade de envio de redefinição de senha, caso haja esquecimento. Neste caso, uma nova senha será enviada para o e-mail cadastrado na plataforma.
- Atualização de Dados na Plataforma: sempre é permitido ao usuário a atualização dos dados de seu perfil do UFollower. Quanto mais informações a plataforma tiver sobre quem a está utilizando, melhores serão as inferências que se poderá fazer. Estes dados atualizados e fidedignos com a situação de quem está usando são preponderantes para recomendações de acessibilidade.
- Monitoramento ubíquo: dependendo do perfil que esteja utilizando a plataforma, um monitoramento pessoal é dirigido. Por exemplo, se um turista estiver entrando em zona de risco, uma notificação será enviada para ele, alertando-o. Caso seja um morador, apenas caso uma situação atípica na localidade em que ele reside aconteça, ele será notificado. Caso contrário, a tendência é de que morador seja notificado menos vezes do que um turista. Um agente de segurança que também tenha o seu Nodo Pessoal cadastrado (um *tablet* em uma viatura, por exemplo) também será notificado. A identificação de situação de risco a algum Nodo Pessoal poderá fazer com que um agente de segurança seja notificado para averiguação. Em caso de aglomerações públicas, caso o mapa de calor da plataforma identifique que em determinada região da cidade está tendo a ocorrência exagerada de determinados fatos, também será um motivo de deslocamento de um agente de segurança previamente notificado pelo UFollower, passando a ele a situação.
- Consulta sobre recursos locais: o agente de recomendação será capaz de inferir pela localidade (Latitude, Longitude) do Nodo Pessoal e desta forma, baseando-se no perfil já cadastrado, sugerir recursos que estejam próximos de onde o Nodo Pessoal está.

4.3.2 Ontologia do modelo UFollower

A ontologia, subcomponente de Processos, conforme pode ser verificado na Figura 21, tem por objetivo gerenciar perfil, realizar recomendação tendo por base o tipo de pessoa, o tipo de ambiente, o tipo de segurança, o tipo de recurso e saúde.

Para que seja possível efetuar um mapeamento das relações existentes em os objetos que configuram a vigilância ubíqua (*Smart Surveillance*) e entre cada uma das características que influenciam nos fatores de segurança para que se possa monitorar e, posteriormente, criar uma da RB utilizadas no processo de monitoramento da condição de violência doméstica é utilizada a ontologia de condições de segurança. Tal ontologia envolve alguns dos conceitos utilizados pelos recursos de segurança. A classe **Pessoa** deve ser utilizada para representar qualquer perfil que possa utilizar o modelo. Um perfil pode ser monitorado por um **TipoRecurso**, que por sua vez poderá estar instalado em um **TipoAmbiente**.

Figura 21: Ontologia de condições de segurança



Fonte: Elaborado pelo Autor.














O **TipoSeguranca** ativada para um **TipoPessoa** poderá ser pública, privada ou *crowdsensing*, onde um grande grupo de indivíduos com algum **TipoRecurso** é capaz de detectar e computar compartilham dados coletivamente e extraem informações para medir, mapear, analisar, estimar ou inferir qualquer processo de segurança de interesse comum.

4.3.3 Sistema Multiagente do modelo UFollower

O componente ARBSVU do modelo UFollower contém o SMA (do inglês *System Management Agent*) responsável pelas ações descritas na seção 4.3.1. O SMA do modelo UFollower, foi desenvolvido na IDE (do inglês *Integrated Development Environment*) Eclipse. Para padronização foi utilizado o *plugin Prometheus Design Tool* (PDT). Os símbolos da notação Prometheus empregados no *plugin* PDT são representados na Figura 22 com a respectiva tradução. A modelagem deste SMA é apresentado na Figura 23.

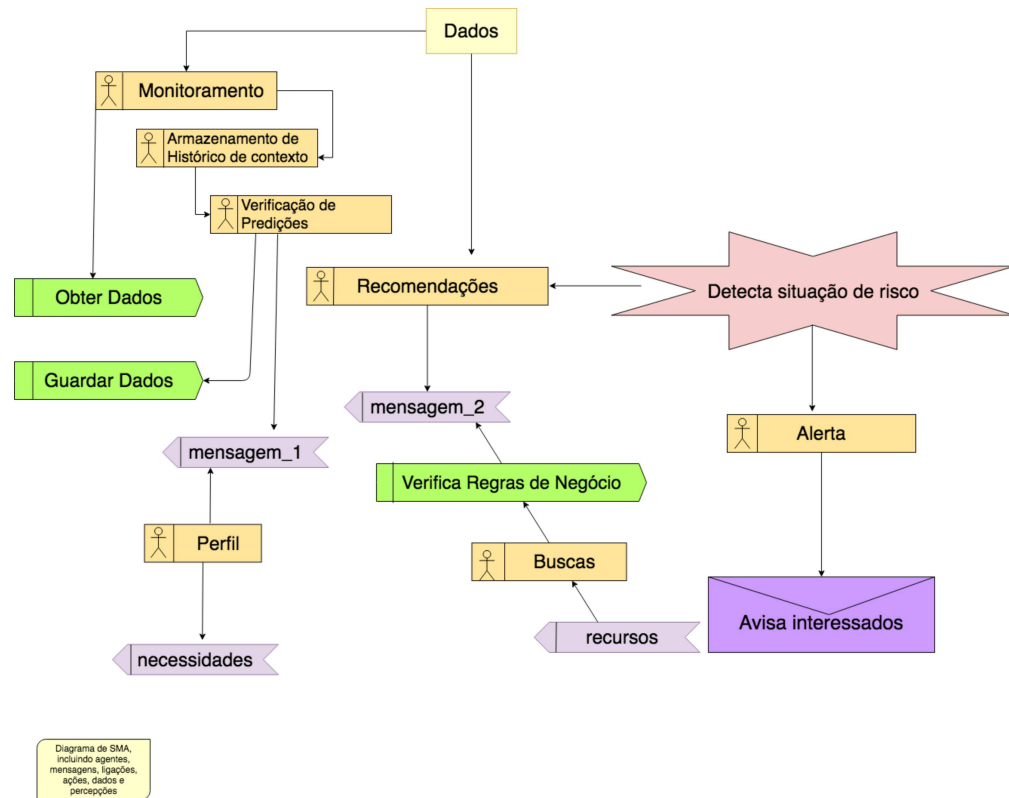
O SMA proposto (ARBSVU) utiliza cinco agentes. O agente representado pela letra “P”, é o agente Perfil, que realiza as operações para gerenciamento do perfil dos usuários do modelo, identificando as necessidades de cada um e os recursos que podem lhe oferecer suporte. O agente representado pela letra “A”, é o agente Alerta, o qual mantém ativa a função de perceber as ações dos usuários e gerar os aletas relativos a essas ações e os dados provindos do ambiente. O agente representado pela letra “M”, é o Monitoramento, que por sua vez, realiza as ações para recebimento e armazenamento de dados e também troca mensagens com o agente Perfil. O agente de Recomendações, simbolizado pela letra “R”, realiza mineração de dados contextualizados e troca mensagens com o agente Buscas, representado pela letra “B”, o qual realiza tarefas para percepção de recursos. O agente Perfil é responsável por gerenciar as preferências dos usuários, enviando as informações coletadas para o agente de monitoramento. Os símbolos da notação do Prometheus são apresentados na Figura 22.

Figura 22- Símbolos da notação Prometheus empregado *plugin* PDT

Notação	Tradução
	NOTE — Anotação
	Actor — Ator
	Agent — Agente
	Action — Ação
	Role — Papel
	Protocol — Protocolo
	Data — Crença / Informação
	Message — Mensagem
	Percept — Percepção
	Scenario — Cenário
	Goal — Objetivo
	Capability — Capacidade
	Plan — Plano

Fonte: Elaborado pelo Autor.

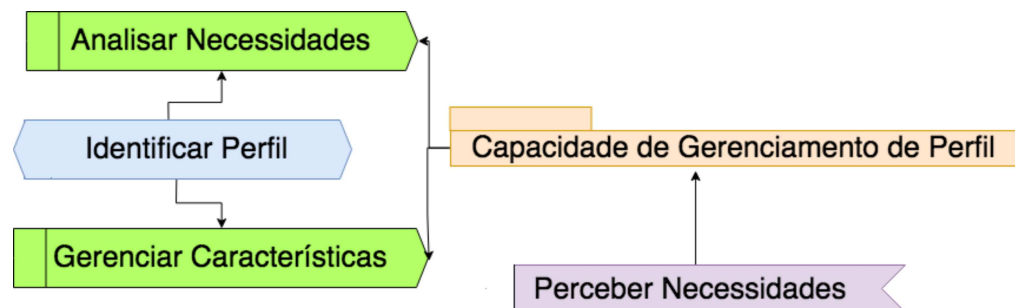
Figura 23: Sistema Multiagente do modelo UFollower



Fonte: Elaborado pelo Autor.

O agente Perfil tem por objetivo tratar das particularidades de cada usuário, fazendo uso de informações tais como suas características pessoais, tipo de usuário (morador, turista, agente de segurança), suas preferências e seu contexto atual. Para isso este agente realiza as ações de verificar necessidades e gerenciar características, a fim de ser capaz de gerenciar o perfil de segurança de cada usuário, conforme pode ser verificado na Figura 24.

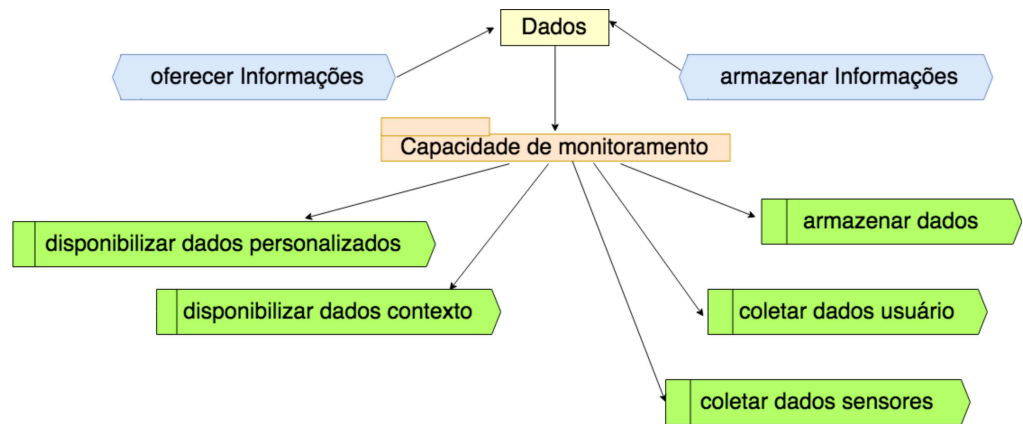
Figura 24: Agente para Perfil do modelo UFollower



Fonte: Elaborado pelo Autor.

O agente Monitoramento recebe e armazena as informações, para oferecer o melhor suporte durante as atividades de segurança ubíqua. Para isso o agente realiza as ações apresentadas na Figura 25.

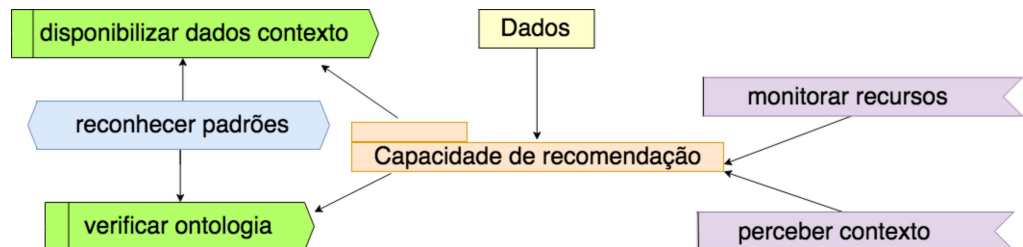
Figura 25: Agente para Monitoramento do modelo UFollower



Fonte: Elaborado pelo Autor.

O agente Monitoramento troca mensagens com o agente de Recomendações e Perfil e compartilha as ações de armazenar e receber dados do agente Buscas. Este é o agente mais importante do modelo UFollower. O agente Recomendações tem sua capacidade e plano, juntamente com suas ações e percepções, apresentados na Figura 26.

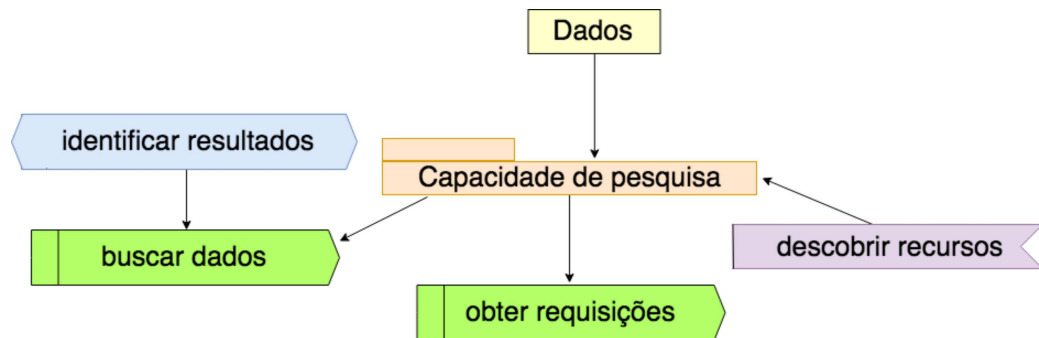
Figura 26: Agente para Recomendação do modelo UFollower



Fonte: Elaborado pelo Autor.

O agente Buscas tem sua capacidade, plano e percepção, juntamente com suas duas ações, apresentados na Figura 27.

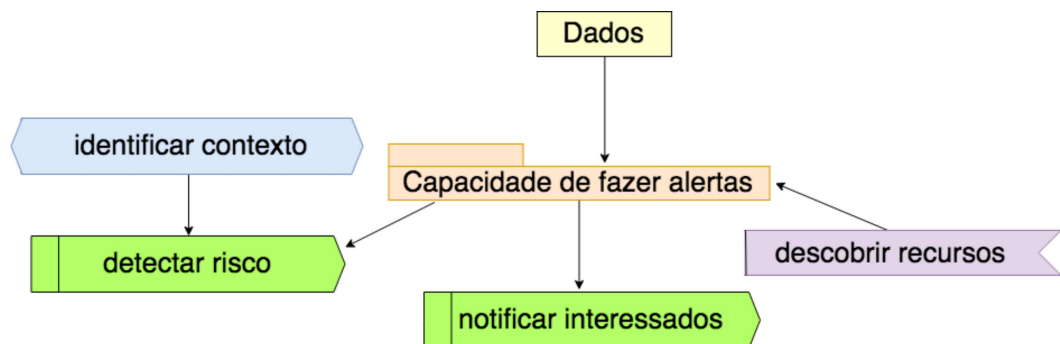
Figura 27: Agente para Buscas do modelo UFollower



Fonte: Elaborado pelo Autor.

O agente para Alertas notifica interessados em receber informações do modelo UFollower. Para isso, ele tem de ser capaz de identificar o contexto no qual o usuário está inserido, descobrir recursos dentro desse contexto, detectar riscos e informar interessados. O agente Alertas tem sua capacidade, plano e percepção, juntamente com suas duas ações, apresentados na Figura 28.

Figura 28: Agente para Alertas do modelo UFollower



Fonte: Elaborado pelo Autor.

O agente de Armazenamento de Histórico de Contexto atua guardando determinadas situações. Um área geográfica, por exemplo, já conhecida pelo agente, quando um usuário de posse de seu *NodoPessoal* passa por ela, ele atua armazenando referências a recursos, pessoas e informações de contexto associadas a esse local.

O agente de Verificações de Predições utiliza como entrada o dados armazenados pelo agente de Armazenamento de Histórico de Contexto. De posse dos dados, esse agente tem a capacidade de prever determinadas situações, atuando de forma preditiva com o intuito de auxiliar o usuário em alguns cenários.

4.3.4 Aplicativo Móvel e Retaguarda *Web* (*Site*)

A seguir, será apresentada a arquitetura para o protótipo e implementação da aplicação servidor, assim como o protótipo para a aplicação cliente.

Para a implementação do protótipo da aplicação servidor, apresentado na Figura 29, será utilizado a linguagem *Python*. Tal linguagem foi escolhida por ser gratuita, multiplataforma e por ser de domínio do autor deste estudo. O servidor *Web* será implementado de forma nativa em *Python* através do módulo `http.server`. Este módulo define classes para implementar servidores HTTP (servidores *Web*) e, embora não seja recomendado para um ambiente de produção por apenas fazer verificações básicas de segurança, para este estudo poderá ser utilizado. Será utilizado o banco de dados MySQL em função de ser robusto, gratuito e de domínio do autor deste estudo.

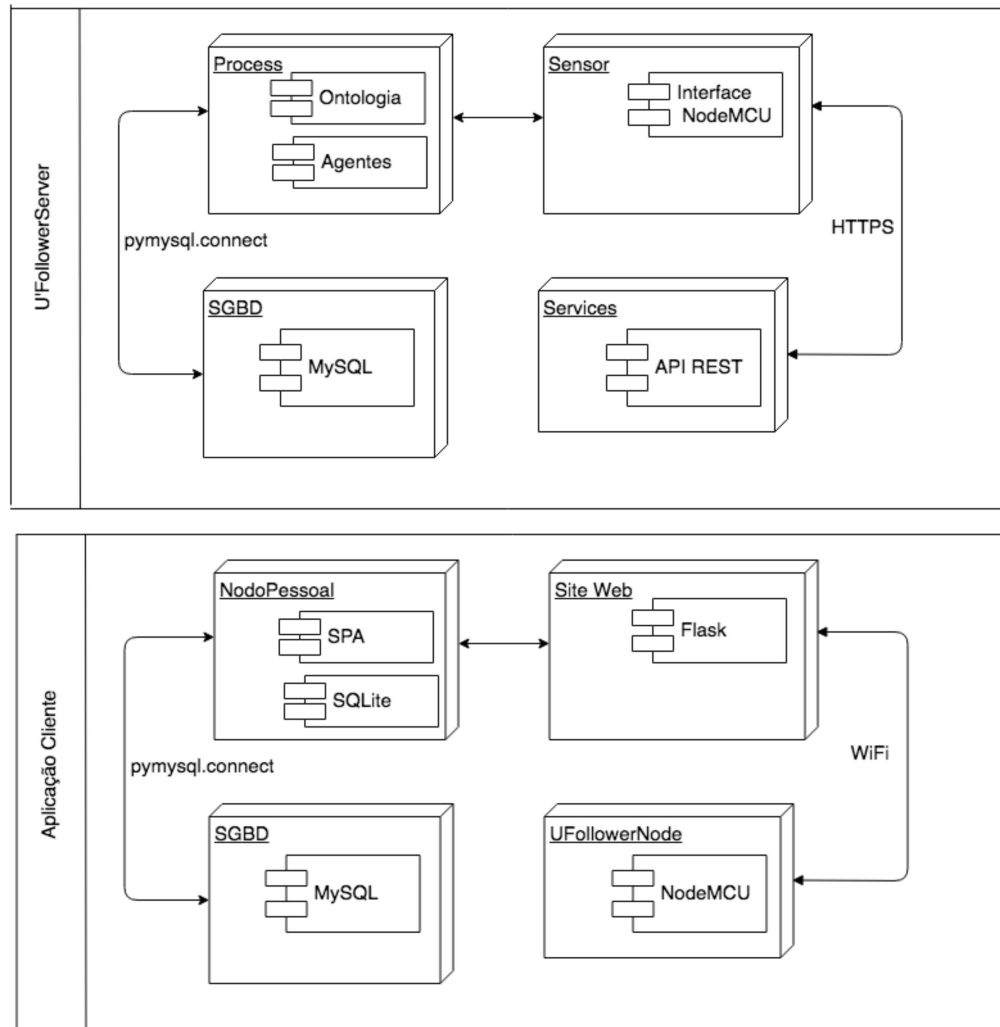
Para o desenvolvimento do protótipo da aplicação cliente, para a parte do *NodoPessoal* será desenvolvida uma *Single Page Application* (SPA). Uma SPA pode criar uma aplicação híbrida que funcione tanto no navegador quanto em dispositivos móveis e *desktop* de forma nativa, podendo o *download* ser executado diretamente das principais lojas de aplicativos (DESHMUKH; MANE; RETAWADE, 2019). Já para a parte *Web*, será desenvolvido um *site* utilizando Flask. Flask é um micro framework em *Python* que fornece a funcionalidade básica da estrutura da *Web* e permite mais *plug-ins* serem adicionados para que a funcionalidade e o conjunto de recursos possam ser estendidos (ARMASH et al. 2019).

A aplicação servidor é responsável pelo repositório de dados do estudo de caso e que atenderá as requisições feitas pela Aplicação Cliente. Sensor no Servidor: o módulo de sensor no servidor será responsável por o processamento dos sinais enviados da aplicação cliente responsável pela coleta de dados nos sensores. Aplicação Cliente: responsável por interagir com o usuário e que enviará requisições para a Aplicação Servidor. Nuvem (*Web*): responsável por disponibilizar um serviço baseado na *Web* para fazer a interligação entre a Aplicação Cliente e Aplicação Servidor.

A comunicação entre as aplicações, tanto a Aplicação Cliente como a Aplicação Servidor, terá como estilo arquitetônico o *Representational State Transfer* (REST), também chamados de serviços *Web* RESTful. Embora o próprio REST não seja um padrão, serviços *Web* RESTful são implementados utilizando padrões *Web*. Cada método *Web* RESTful é representado por um URL único. A escolha por este tipo de serviço *Web* é por não estar limitado a retornar dados num determinado formato de arquivo (DEITEL, 2010).

A Figura 29 ilustra como foi projetada a arquitetura do UFollower.

Figura 29: Arquitetura do Aplicativo Móvel e do *Site Web* do modelo UFollower



Fonte: Elaborado pelo Autor.

4.4 Considerações sobre o capítulo

Ao longo deste capítulo foram detalhados os critérios sobre tecnologias do modelo UFollower. Além disso o modelo foi apresentado e a arquitetura foi definida. O SMA do modelo UFollower foi apresentado para definir o funcionamento de seu mecanismo de monitoramento dos usuários e seu sistema de inferências que se baseia no perfil, contexto e ambientes atendidos pelo modelo. Os requisitos do modelo foram introduzidos para que no próximo capítulo a implementação de um protótipo seja realizada e avaliada.

5 ASPECTOS DE IMPLEMENTAÇÃO E AVALIAÇÃO

Este capítulo apresenta aspectos de implementação e avaliação do modelo UFollower. Para realizar a avaliação, será desenvolvido um protótipo do modelo contemplando o servidor UFollowerServer, o *Site Web*, o assistente pessoal NodoPessoal e sensores UFollowerNode. Para verificar o suporte oferecido pelo protótipo, o mesmo será integrado a sistemas existentes tomando como base dados abertos do estado do Rio Grande do Sul. Os dados sobre índices de concentração de determinados fatos de ocorrências foram coletados através de material da Secretaria de Segurança Pública do Estado. O texto do presente capítulo apresenta na primeira seção aspectos relacionados a implementação do protótipo. Na segunda seção é apresentada a metodologia de avaliação.

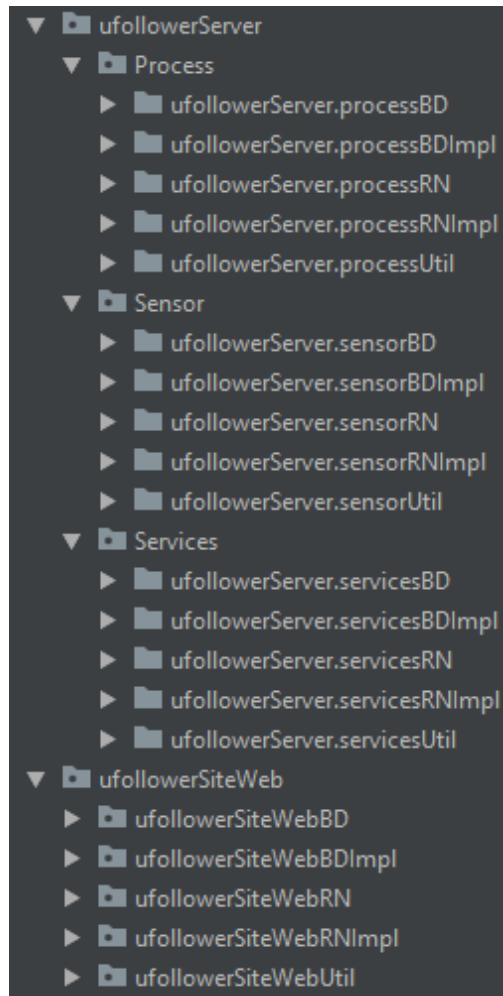
5.1 Implementação do protótipo

O protótipo do modelo é composto por quatro itens, UFollowerServer, *Site Web*, NodoPessoal e UFollowerNode. A seguir, os mesmos são apresentados. As classes foram geradas em um mesmo projeto, mas serão segmentadas em projetos distintos. O UFollowerServer utilizará as classes geradas pelo componentServer e será configurado na computação em nuvem, possuindo basicamente Web services, agentes, ontologia e bases de dados. O *site Web* utilizará as classes geradas pelo componentWeb, utilizando Python e pode ser hospedado em servidores de páginas *Web*. O NodoPessoal utilizará as classes geradas pelo componentMobile e o mesmo será desenvolvido em Angular. Angular é um modelo *front-end* com o qual é possível criar aplicativos com reutilização de código destinados a qualquer tipo de implantação: *Web*, móvel nativo e *desktop* nativo. Angular permite estender a linguagem do modelo com componentes próprios e usar uma grande variedade de componentes existentes. Desde o protótipo até a implantação global, o Angular fornece a produtividade e a infraestrutura escalável que suporta os maiores aplicativos do mundo (ANGULAR, 2019).

5.1.1 Implementação do Servidor UFollower

O UFollowerServer foi desenvolvido com recursos de programação de *Python 3* e *Connexion* para criar APIs REST úteis que poderão incluir validação de entrada e saída e a documentação do modelo serão fornecida pelo Swagger. O Swagger é um conjunto de ferramentas de código aberto criadas em torno da especificação OpenAPI que auxiliam a projetar, criar, documentar e consumir API REST (SWAGGER, 2019). O servidor será composto por *Web services* que utilizam a implementação RESTfull *Web Services* os quais atendem um conjunto de chamadas que se concentram em interfaces. O UFollowerServer será dividido em três projetos que se comunicam entre si: *Process*, *Sensor* e *Services*. Cada um desses projetos adota o padrão arquitetura MVC - *Model View Control*, sendo que a parte *View* é apresentada pelo outro projeto chamado *ufollowerSiteWeb*. Os projetos BD expõem as interfaces de banco de dados e os projetos BDImpl apresentam as implementações dessas interfaces. Essa mesma abordagem vale para os três projetos que compõem o *ufollowerServer*. Os projetos Util têm classes utilitárias que são utilizadas no contexto ao qual estão inseridas. A Figura 30 ilustra os pacotes criados para a implementação dessa parte do protótipo.

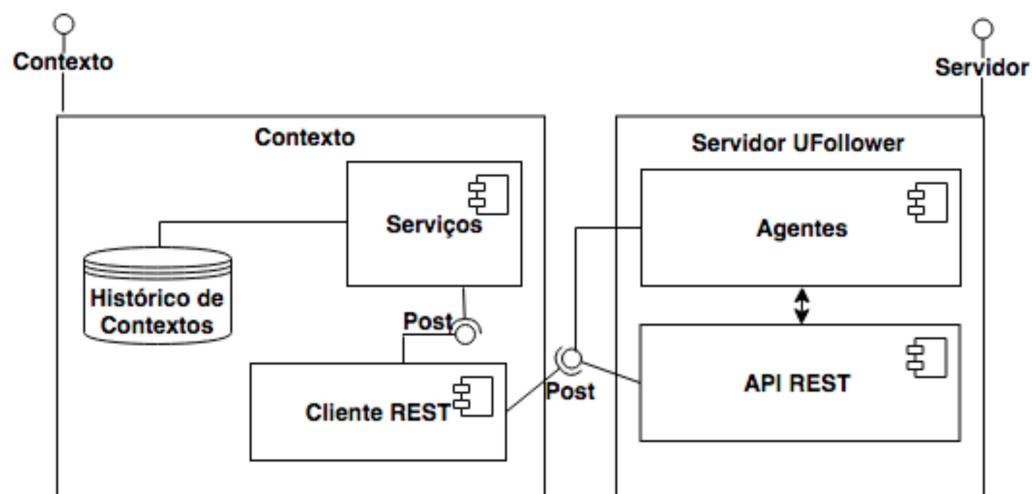
Figura 30: Pacotes básicos do protótipo implementado



Fonte: Elaborado pelo Autor.

A Figura 31 representa a comunicação do contexto com o servidor. O contexto representado por seus serviços, histórico de contexto e cliente REST, comunica-se com o servidor utilizando o método de requisição POST. Esse método de requisição foi utilizado para solicitar que o servidor *Web* aceite os dados anexados no corpo da mensagem de requisição para armazenamento. No lado do servidor, ao receber a requisição seus agentes atuam e a resposta é devolvida para o cliente o qual a consumirá através da API REST fornecida. Conforme esses artefatos vão sendo utilizando, serão melhor explorados e explicados no decorrer das próximas seções.

Figura 31: Representação do contexto interagindo com o Servidor



Fonte: Elaborado pelo Autor.

5.1.2 Implementação dos Clientes

Com relação aos clientes, serão desenvolvidos três níveis: Cliente *Site Web*, Cliente Nodo Pessoal e Cliente Sensor UFollowerNode. A implementação do *site Web* será feita em Flask e não é vinculada ao UFollowerServer, podendo ser hospedada em outro local. Visando obedecer futuras migrações ou ampliações na disponibilização dos serviços principais, para o UFollowerServer serão conceitos de computação em nuvem, tais como elasticidade e disponibilidade. A implantação do Nodo Pessoal, será por meio de aplicativo *single-page application* em Angular e será executada em dispositivos tais como smartphones e *tablets*. O aplicativo utilizará ferramentas para exibição de mapas e armazenamento de dados locais baseado em dados abertos, o *OpenStreetMap* (OSM, 2015). A implementação do Cliente Sensor UFollowerNode pode ser desenvolvida em diversos dispositivos, tais como os microprocessadores nodeMCU, Raspberry Pi, Nanode, PCduino, Waspote ou mesmo o microcontrolador Arduino. Todos estes dispositivos suportam conexão Wi-Fi, 3G, módulo RFID (do inglês *Radio-Frequency Identification*) e podem ser incrementados com sensores.

5.2 Metodologia e aspectos de avaliação

De forma a responder à questão de pesquisa, foi desenvolvida uma avaliação do modelo, aplicando-o em 3 cenários que foram avaliados através de simulação:

- Cenário 1: Com a utilização de 2000 registros de dados realísticos sobre ocorrências de agressão doméstica, gerados pela ferramenta Mockaroo, foram aplicadas Redes Bayesianas para a inferência do fato em questão. Como os dados foram baseados em ocorrências reais, sabe-se que algumas premissas devem ser conhecidas para permitir a inferência: como se é ou não a primeira agressão que a vítima sofreu, se ela chegou a solicitar medida protetiva contra o agressor, se chegou a ter a possibilidade de fazer uma nova denúncia e, por fim, a inferência em cima do fato ter sido consumado ou não. Para isso foram mapeados os registros em 4 colunas: *primeira_agressao*, *medida_protetiva*,

nova_denuncia e consumado. Dada as 3 primeiras colunas, procurou-se prever a taxa de acerto da última coluna.

- Cenário 2: Simulação de cenário em que um morador monitorado por tornozeleira eletrônica está infringindo medida legal. Ele já agrediu uma moradora e deve manter uma distância de, no mínimo, 500m dela.
- Cenário 3: Simulação de cenário em que uma turista dos Estados Unidos da América está deslocando pela capital do Estado do Rio Grande do Sul. Durante o seu trajeto é realizada a consulta no histórico de contexto do bairro onde ela está para verificar o grau de periculosidade da região. Em um dado momento ela está prestes a entrar em um bairro historicamente perigoso.

Para avaliação do protótipo foi realizada uma simulação dos cenários compostos por ambientes, recursos e usuários. Os ambientes serão dotados de recursos dinâmicos e estáticos. O cenário para avaliação será uma cidade de Porto Alegre - RS. Utilizando-se dos dados realísticos extraídos da plataforma Mockaroo, foram mapeados os pontos onde há mais incidência de crimes, com o intuito de auxiliar os três perfis do modelo: morador, turista e agente de segurança.

A avaliação do modelo foi realizada em um ambiente simulado utilizando a linguagem *Python* para a geração de *scripts*, a plataforma OpenStreetMap para a geração de mapas e um banco de dados relacional para o armazenamento dos dados. Conforme pode ser visto na Figura 32, o *dataframe* utilizado para o cenário 1 foi composto de 2000 linhas e 4 colunas.

Figura 32: Quantidade de Registros para inferência de Redes Bayesianas

Quantidade de Registros produzidos pelo Mockaroo

```
import pandas as pd

dataframe = pd.read_csv("dados_feminicidio.csv")
dataframe.shape
```

(2000, 4)

Fonte: Elaborada pelo Autor.

Para a avaliação do ambiente simulado foram utilizados 20 objetos, divididos em 12 nodos pessoais, 5 câmeras de monitoramento instaladas em vias públicas e 3 veículos, conforme pode ser verificado nas Tabelas 3, 4 e 5. Os 12 nodos pessoais foram divididos entre 6 moradores, 3 turistas e 3 agentes. Entre os moradores, 3 homens e 3 mulheres e entre as mulheres, há 2 com medida protetiva, por violência doméstica, contra dois homens da simulação. Todos os 12 nodos pessoais têm sua localização compartilhada com o modelo pela GPS de seus dispositivos. Além disso, os 2 homens para os quais há medida protetiva contra eles também compartilham sua localização pela sinal de GPS de suas tornozeleiras.

Consta ainda como parte da simulação o fato de que um dos 3 turistas estar se dirigindo a um bairro considerado perigoso na cidade de Porto Alegre. Todos os conjuntos de dados representam uma amostra de uma quantidade superior existente em uma cidade real, tendo a sua utilização apenas para servir como prova de conceito do modelo proposto. Assim, é possível verificar que a Tabela 3 ilustra um universo rico para fins de simulação do UFollower. Há dois

moradores monitorados tanto pelo nodo pessoal como pelas suas tornozeleiras. Há também mulheres com medida protetiva contra dois homens da simulação.

Tabela 3: Nodos Pessoais utilizados na simulação do modelo UFollower

<i>Id</i>	<i>Perfil</i>	<i>Nome</i>	<i>Gênero</i>	<i>Lat</i>	<i>Lon</i>	<i>Med_Prot_Id</i>	<i>Tornozeleira_Id</i>
1	Morador	Morador_1	F	-30.027336	-51.175452	2	-
2	Morador	Morador_2	M	-30.020953	-51.17528	-	TORNOZELEIR A001
3	Morador	Morador_3	F	-30.027336	-51.175452	-	-
4	Turista	Turista_1	F	-30.001656	-51.091359	-	-
5	Agente	Agente_1	M	-30.028848	-51.192653	-	-
6	Agente	Agente_2	M	-30.027176	-51.206365	-	-
7	Turista	Turista_2	M	-30.034505	-51.231191	-	-
8	Morador	Morador_4	M	-29.921151	-51.140146	-	TORNOZELEIR A002
9	Agente	Agente_3	F	-30.056863	-51.192124	-	-
10	Turista	Turista_3	M	-30.134919	-51.231713	-	-
11	Morador	Morador_5	F	-30.020037	-51.179350	8	-
12	Morador	Morador_6	M	-30.020037	-51.179350	-	-

Fonte: Elaborado pelo Autor.

Na Tabela 4 estão as 5 câmeras utilizadas na simulação. As câmeras na simulação têm a função de identificar veículos e consumir a API REST do modelo UFollower para saber a situação daquele veículo. Entre as situações possíveis, constam: indicativo de furto e roubo, clonagem e averiguação de proprietário, para identificar algum indicativo- crime ativo, por exemplo, descumprimento de alguma medida legal.

Para a simulação, foi deixado um veículo entrar em zona geográfica com distanciamento inferior a 500m de uma mulher vítima de violência doméstica do proprietário do veículo em questão. Admitindo-se que o condutor do veículo pesquisado seja quem o esteja conduzindo, quando câmeras da simulação identificaram seu veículo, o modelo enviou alertas, através dos Agentes de Recomendação e Recursos Locais.

Tabela 4: Câmeras utilizadas na simulação do modelo UFollower

<i>Id</i>	<i>Lon</i>	<i>Lon</i>	<i>Nome</i>	<i>Endereço</i>
1	-30.027336	-51.175452	Câmera 1	Rua Anita Garibaldi
2	-30.020953	-51.173528	Câmera 2	Rua Tomaz Gonzaga
3	-30.027336	-51.175452	Câmera 3	Rua 14 de Julho
4	-30.001656	-51.091359	Câmera 4	Avenida Plínio Brasil Milano
5	-30.028848	-51.192653	Câmera 5	Rua 24 de Outubro

Fonte: Elaborado pelo Autor.

Na Tabela 5 estão os 3 veículos utilizados para a simulação. Além da identificação da placa do veículo, há uma referência ao proprietário cadastrado na base de dados. A informação *id_nodo_pessoal* se refere ao id da Tabela 3. Desta forma é possível saber, por exemplo, que o veículo de placas IZA-8J32, na simulação, pertence ao *Morador_2*. Além dessa informação, uma vez tendo-se o apontamento correto para a tabela de nodos pessoais, é possível verificar também que para esse morador há uma tonozeleira vinculada (TORNOZELEIRA001).

Tabela 5: Veículos utilizados na simulação do modelo UFollower

<i>Id</i>	<i>Placa</i>	<i>Proprietário</i>
1	IZA-8J32	{id_nodo_pessoal: 2}
2	IJK-0A23	{id_nodo_pessoal: 8}
3	IFF-9C00	{id_nodo_pessoal: 11}

Fonte: Elaborado pelo Autor.

Para melhor entendimento das avaliações as mesmas serão apresentadas nas seções e subseções a seguir. Através deste conjunto de avaliações pretende-se responder à questão de pesquisa do trabalho.

Para a avaliação de aplicações ubíquas e sistemas sensíveis ao contexto, uma das alternativas é a utilização de cenários (BARBOSA et al., 2018; CARPENTIERI et al., 2020). Fazendo uso dessa estratégia, foram criados 2 cenários simulados para a avaliação do UFollower. No primeiro cenário foi simulado um caso de agressão doméstica com possível ocorrência futura de feminicídio. No segundo cenário, foi simulado um passeio na cidade de Porto Alegre por uma turista dos EUA. Para fins de contextualização da avaliação de cada cenário, serão, primeiramente, apresentado dados que corroboram com suas simulações.

5.2.1 Cenário 1

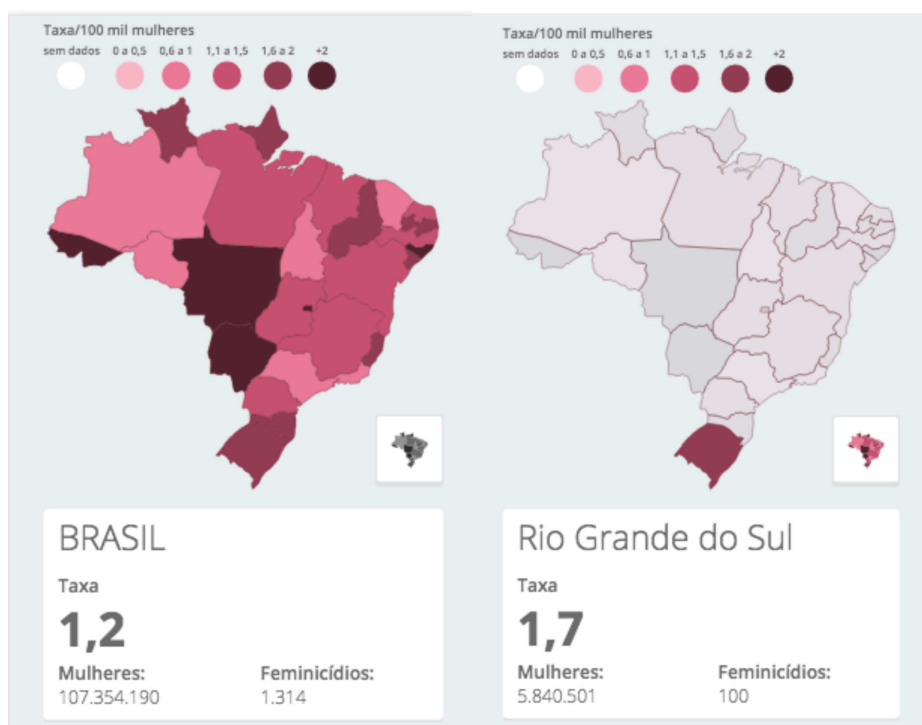
A violência contra a mulher não é um fato recente, desde os primórdios da humanidade as mulheres vêm sendo vítimas de agressões, muitas vezes chegando a óbito. Por outro lado, o

que é novo, é a responsabilidade de vencer tal violência, como condição para a construção da humanidade, visto que o feminicídio define-se como a expressão máxima da violência contra a mulher (FERNANDA et al., 2018).

Tal violência ocorre em várias esferas da vida e se manifesta sob formas e circunstâncias distintas. Neste contexto, dentre as inúmeras situações de violência que vitimam as mulheres, destacam-se, às ocorridas no espaço definido socialmente para as mulheres: o espaço privado, a família e o domicílio (SANTI, 2010).

Conforme pode ser verificado na Figura 33, a taxa de feminicídio no Estado do Rio Grande do Sul é maior do que a taxa nacional. A Figura 34 ilustra percentuais sobre quais dias da semana o crime mais acontece, se ocorre durante o dia, quanto acontece em área urbana, entre outros percentuais.

Figura 33: Mapa do Feminicídio



Fonte: LIMA (2020).

Figura 34: Observatório da Violência contra a Mulher da SSP-RS

Mapa dos feminicídios no RS

Em 2019, o Rio Grande do Sul registrou 91 feminicídios*



Fonte: SSP (2020).

Entre os índices exibidos pela Figura 34, um deles chama mais a atenção: 71% das vítimas foram mortas. E o *modus operandi* foi a utilização de facas ou tiros. Esse percentual, em particular, permite-nos uma reflexão: e se um modelo pudesse, através do relato da vítima, inferir se ela é um ponto potencial de estar nos 71% dessa estatística, ou não.

Desta forma, avaliou-se a base de dados do Estado do Rio Grande do Sul, mais especificamente as tabelas de Ocorrência, Fato, Participantes e *Modus Operandi* e foi identificado que as informações primeira_agressao, medida_protetiva e nova_denuncia da tabela 6 são preponderantes para um modelo baseado em Redes Bayesianas inferir sobre a coluna consumado. Na tabela, caso a informação esteja preenchida com 0 significa a negativa do nome da coluna. Ou seja, o valor 0 na coluna primeira_agressão significa para o modelo que não se trata da primeira_agressão da vítima. Caso essa coluna esteja com o valor 1, é a primeira agressão da vítima. A Tabela 6 apresenta a amostra de dados tabelados para as Redes Bayesianas.

Tabela 6: Amostra de Dados tabelados para o modelo RB

<i>consumado</i>	<i>primeira_ agressão</i>	<i>medida_ protetiva</i>	<i>nova_ denuncia</i>
1	0	1	0
1	0	1	1
0	0	0	0
1	1	1	0
0	1	1	0
0	1	1	0
1	1	1	1
1	0	1	0
1	1	1	1

Fonte: Elaborado pelo Autor.

A Tabela 6 tem, originalmente, 2.000 registros policiais envolvendo vítima mulher. Para esse *dataset* foram utilizados os algoritmos apresentados na Tabela 7.

Tabela 7: Algoritmos baseados em Naive Bayes

<i>Algoritmo</i>	<i>Característica</i>
<i>MultinomialNB</i>	A distribuição é parametrizada por vetores para cada classe, onde está o número de características (na classificação do texto, o tamanho do vocabulário) e a probabilidade do recurso que aparece em uma amostra pertencente à classe.
<i>GaussianNB</i>	implementa o algoritmo gaussiano Naive Bayes para classificação. A probabilidade dos recursos é assumida como gaussiana.
<i>ComplementNB</i>	é uma adaptação do algoritmo multinomial padrão, que é particularmente adequado para conjuntos de dados desequilibrados. Especificamente, a este algoritmo usa estatísticas do complemento de cada classe para calcular os pesos do modelo.
<i>BernoulliNB</i>	Essa classe exige que as amostras sejam representadas como vetores de recursos com valor binário; se entregue qualquer outro tipo de dados, uma instância do BernoulliNB pode tornar binária essa entrada.
<i>CategoricalNB</i>	Este algoritmo pressupõe que cada recurso, descrito pelo índice, tenha sua própria distribuição categórica.

Fonte: Elaborado pelo Autor.

Figura 35: Trecho da implementação dos modelos NB do Scikit Learn

```

from sklearn.naive_bayes import MultinomialNB
modeloMultinomial = MultinomialNB()
resultadoMultinomial = fit_and_predict("MultinomialNB", modeloMultinomial, treino_dados, treino_marcacoes, teste_dados,
                                       teste_marcacoes)

from sklearn.ensemble import AdaBoostClassifier
modeloAdaBoost = AdaBoostClassifier()
resultadoAdaBoost = fit_and_predict("AdaBoostClassifier", modeloAdaBoost, treino_dados, treino_marcacoes, teste_dados,
                                   teste_marcacoes)

from sklearn import svm
modeloSVM = svm.SVC(kernel='linear')
resultadoSvmClassifier = fit_and_predict("SVM", modeloSVM, treino_dados, treino_marcacoes, teste_dados,
                                       teste_marcacoes)

from sklearn.naive_bayes import GaussianNB
modeloGaussian = GaussianNB()
resultadoGaussianNB = fit_and_predict("GaussianNB", modeloGaussian, treino_dados, treino_marcacoes, teste_dados,
                                      teste_marcacoes)

```

Fonte: Elaborado pelo Autor.

O código completo, com todas as implementações exibidas em partes na Figura 35 foi desenvolvido utilizando o Ambiente de Desenvolvimento Integrado (IDE - *Integrated Development Environment*) PyCharm. O PyCharm fornece preenchimento inteligente de código, inspeções de código, realce rápido de erros e correções rápidas, juntamente com refatorações de código automatizadas e recursos avançados de navegação (HU, 2018).

Dos 2000 registros, 80% foram utilizados para o treinamento do modelo e 20% foram utilizados para os testes. A taxa de acerto de todos ficou 70.35%. Para validar os algoritmos, foi desenvolvido um algoritmo base que sempre infere o valor 1 para a coluna consumado, independente dos valores das colunas primeira_agressao, medida_protetiva e nova_denuncia. Nessa situação, o algoritmo base acertou 66,17% das vezes.

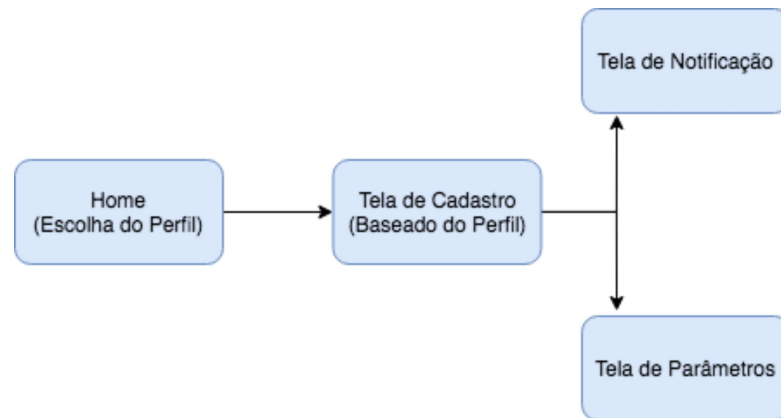
Desta forma, exemplificando em um caso hipotético, caso uma mulher se dirija a um balcão de uma delegacia de polícia relatando agressão doméstica, ao lavrar o histórico da ocorrência já estará sendo analisado o seu caso pelo modelo. Se o este inferir os 70.35%, a autoridade policial, na figura de um delegado de polícia, poderá agir sobre esse dado, reforçando a segurança da vítima, agindo de forma preventiva. Desta forma, mesmo utilizando a implementação de uma Rede Bayesiana disponível na biblioteca de aprendizado de máquina Scikit-learn, comparando-se com os trabalhos relacionados, o modelo UFollower foi o único que utilizou tal abordagem.

5.2.2 Cenário 2

O segundo cenário diz respeito a utilização de simulação para representar o fato de um homem ter de ficar mais de 500m de uma mulher detentora de medida protetiva. A mulher em questão figura entre as vítimas do cenário 1, no entanto, não teve o fato consumado, ou seja, houve violência doméstica por parte desse homem, mas não o feminicídio em si.

Primeiramente, será apresentado como essa mulher fez para utilizar o UFollower App. A Figura 36 representa o fluxo necessário para se utilizar o aplicativo.

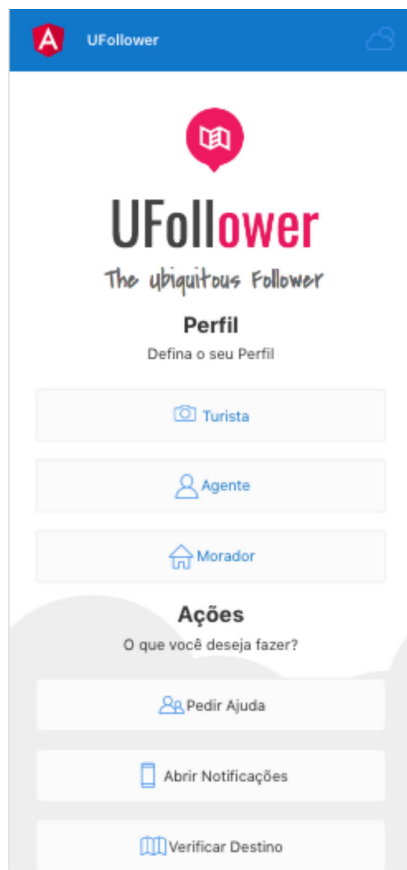
Figura 36 - Fluxo do uso do UFollower App



Fonte: Elaborado pelo Autor.

A Figura 37 representa a interface principal do UFollower App. Nessa interface, o usuário escolherá qual será o seu perfil no modelo. No caso, a mulher da simulação escolheu o perfil Morador. A Figura 38 ilustra os dados que requerem preenchimento para o perfil Morador. Com o objetivo de manter a interface somente com os dados necessários, validações são feitas quando pelo *backend* quando os dados são submetidos.

Figura 37: Aplicativo UFollower Nodo Pessoal



Fonte: Elaborado pelo Autor.

Figura 38: Perfil Morador no App



Fonte: Elaborado pelo Autor.

A Figura 39 ilustra a retaguarda *Web* do modelo UFollower. O papel desse módulo no modelo é servir de *interface* entre o Perfil Agente e os demais perfis. Na nodo *Web* toda a informação que foi cadastrada no Nodo Pessoal pode ser visualizada, editada e deletada. Desta forma é gerado um histórico de contexto, tanto do perfil do usuário quanto das informações geográficas capturadas pelo GPS do Nodo Pessoal o qual esteja portando.

Figura 39: Módulo *Web* do Modelo UFollower

U'Follower Web

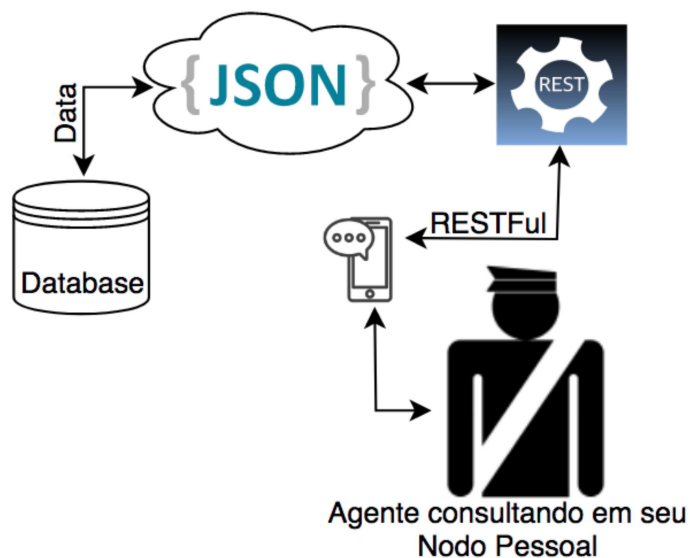
Administração			
Nome	Perfil	Observação	Ações
Morador_1	Morador	Ex-Companheiro, Morador_2, em liberdade Condicional, tornozeleira	Editar Deletar Listar
Morador_2	Morador	Indiciado em fato relacionado à violência doméstica ou familiar contra a mulher	Editar Deletar Listar
Turista_1	Turista	Tursita dos Estados Unidos da América	Editar Deletar Listar
Agente	Agente	Agente da Segurança do Estado do Rio Grande do Sul	Editar Deletar Listar
Agente_2	Agente	Atendente do Serviço de Atendimento Móvel de Urgência	Editar Deletar Listar
Turista_2	Turista	Morador do Estado do Rio de Janeiro	Editar Deletar Listar

Fonte: Elaborado pelo Autor

Em virtude de haver muitos tipos de dados na maneira como o modelo foi estruturado, a maneira adotada para realizar o tráfego desses dados entre as diversas camadas do UFollower foi utilizando o padrão JSON (*Javascript Object Notation*) *Schema* disponibilizado no formato de uma API (*Application Programming Interface*) REST (*Representational State Transfer*). Para Frozza (2018), um valor JSON pode ser um valor de tipo primitivo (*string*, numérico, booleano), um tipo estruturado (objeto, matriz) ou nulo. Ao usar objetos e matrizes como valores, é possível definir documentos JSON com estruturas aninhadas.

A Figura 40 ilustra o comportamento de consumo da API em um nodo pessoal. No exemplo, um Agente de Segurança Pública, um dos perfis do UFollower App, está utilizando o seu nodo pessoal para consultar algo no aplicativo. O dado vem em sua forma estruturada do banco de dados, é convertido para o formato JSON e é disponibilizado para o cliente final através de uma API REST.

Figura 40: Representação de consumo da API REST no nodo pessoal



Fonte: Elaborado pelo Autor.

Para proporcionar a evolução futura do modelo, foi desenvolvida uma API para consumo de recursos do UFollower. A Figura 41 ilustra parte dessa API, com todas as possibilidades de consumo de serviços destinados aos Nodos Pessoais.

Figura 41: API completa para utilização do UFollower

API UFollower ^{1.0}
 [Base URL: /]
<http://127.0.0.1:5000/swagger.json>

API UFollower

Operações: Nodo Pessoal

POST	/Operações: Nodo Pessoal/	Criar um Recurso Nodo Pessoal
GET	/Operações: Nodo Pessoal/	Listar Nodos
PUT	/Operações: Nodo Pessoal/{id}	Atualizar um Recurso Nodo Pessoal através do seu identificador
GET	/Operações: Nodo Pessoal/{id}	Buscar um Recurso Nodo Pessoal
DELETE	/Operações: Nodo Pessoal/{id}	Deletar um Recurso Nodo Pessoal através do seu identificador

Fonte: Elaborado pelo Autor.

As Figura 42 e 43 ilustram exemplos reais de dados estruturados do modelo sendo disponibilizado via API para consumo no formato JSON. Na Figura 42, foi solicitado à API o perfil 1. O retorno foi uma mulher, de 34 anos, brasileira, que já foi agredida pelo ex-companheiro, o qual está solto sendo monitorado por tornozeleira. A Figura 43 ilustra a solicitação feita à API do perfil 2. Nesse caso, o retorno foi um homem, agressor que está sendo monitorado por já ter sido indiciado em fato relativo violência doméstica.

Figura 42: API disponibilizada pela retaguarda *Web* do modelo UFollower(a)



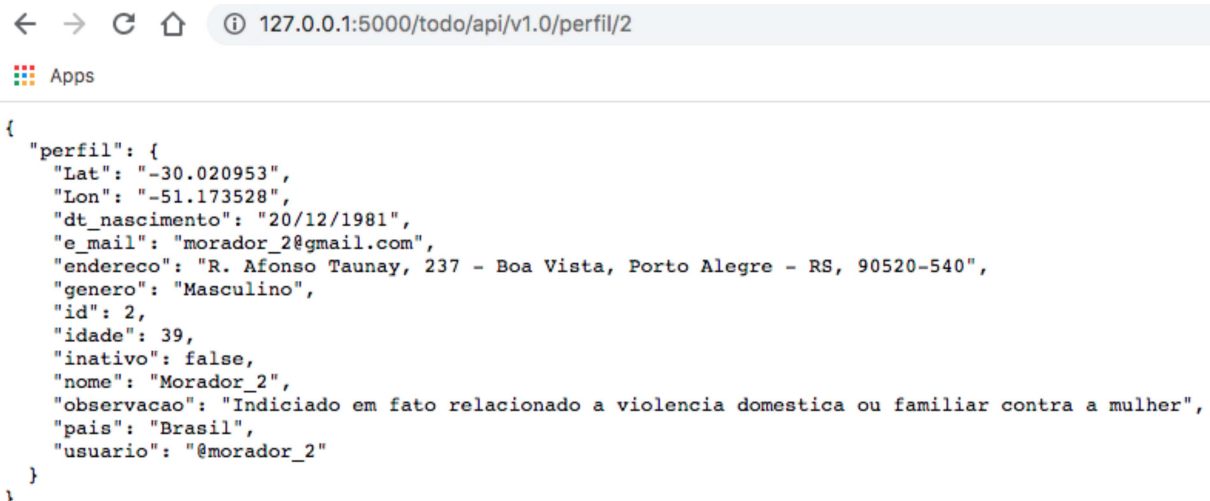
```

{
  "perfil": {
    "Lat": "-30.027336",
    "Lon": "-51.175452",
    "dt_nascimento": "20/12/1986",
    "e_mail": "morador_1@gmail.com",
    "endereco": "Av. Dr. Nilo Pecanha, 1600 - Boa Vista, Porto Alegre - RS, 91330-002",
    "genero": "Feminino",
    "id": 1,
    "idade": 34,
    "inativo": false,
    "nome": "Morador_1",
    "observacao": "Ex-Companheiro, Morador_2, em liberdade condicional, tornozeleira",
    "pais": "Brasil",
    "usuario": "@morador_1"
  }
}

```

Fonte: Elaborado pelo Autor.

Figura 43: API disponibilizada pela retaguarda *Web* do modelo UFollower(b)



```

{
  "perfil": {
    "Lat": "-30.020953",
    "Lon": "-51.173528",
    "dt_nascimento": "20/12/1981",
    "e_mail": "morador_2@gmail.com",
    "endereco": "R. Afonso Taunay, 237 - Boa Vista, Porto Alegre - RS, 90520-540",
    "genero": "Masculino",
    "id": 2,
    "idade": 39,
    "inativo": false,
    "nome": "Morador_2",
    "observacao": "Indiciado em fato relacionado a violencia domestica ou familiar contra a mulher",
    "pais": "Brasil",
    "usuario": "@morador_2"
  }
}

```

Fonte: Elaborado pelo Autor.

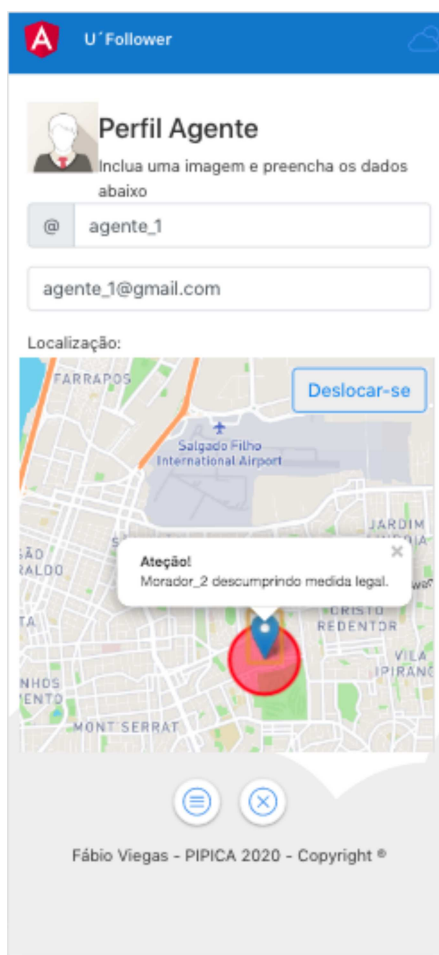
Desta maneira, é possível a um usuário com o perfil de Agente ser notificado, caso uma haja algum alerta. O georreferenciamento é conquistado através do GPS do Nodo Pessoal do perfil morador_1 e do Nodo Pessoal e Tornozeleira do perfil mordador_2. Assim, mapas

interativos notificam o perfil destino daquela informação. Os mapas interativos no UFollower são fornecidos utilizando *Leaflet*. O *Leaflet* é uma biblioteca escrita em *JavaScript*, muito útil para criar mapas interativos e fáceis de usar. Suas principais vantagens são que ele é de código aberto e permite o uso de uma API que facilita a implementação de ferramentas de simulação e outros aplicativos (TAYEBI, 2019).

A partir do momento que o perfil Agente é notificado sobre um Alerta, ele pode deslocar-se até lá, utilizando a API do *Leaflet* que traça rotas caminhando, de bicicleta ou de automóvel.

Assim como a moradora da simulação escolheu na Figura 36 para qual perfil o modelo lhe atenderá, o Agente_1 também o fez, escolhendo o perfil Agente. A Figura 44 ilustra a interface do perfil Agente sendo notificada de um descumprimento de medida legal.

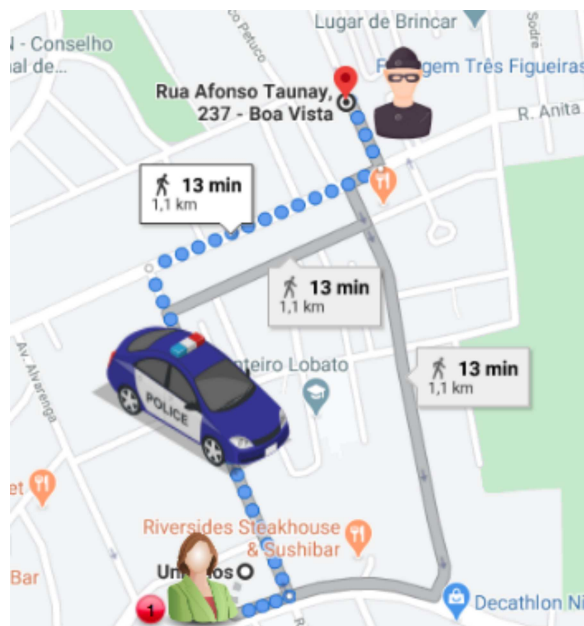
Figura 44: Perfil Agente sendo notificado em seu Nodo Pessoal



Fonte: Elaborado pelo Autor.

De posse dos dados da notificação, espera-se que o Agente possa interceptar o Morador agressor o quanto antes, não permitindo que ele se aproxime da Moradora que tem medida protetiva contra ele. A Figura 45 ilustra essa intenção, caso o morador se desloque a pé até o local de trabalho da Moradora.

Figura 45: Morador_2 caminhando sendo interceptado pelo Agente antes de se aproximar mais do Morador_1



Fonte: Elaborado pelo Autor.

O deslocamento proporcionado pelo botão “Deslocar-se” apresentado na Figura 43, calculou a melhor rota para a viatura do Agente chegar até a posição estratégica ilustrada na Figura 39. Para fins de exemplificação e construindo um cenário hipotético, a mulher de casaco verde na Figura 44 (Morador_1) tem medida protetiva a qual exige que seu ex-companheiro (Morador_2), identificado pelo homem de preto, permaneça, no mínimo, 500m afastado de seu local de trabalho, identificado pelo número 1 no círculo vermelho. Conforme pode ser verificado, o Morador_2 está descumprindo a medida protetiva, deslocando-se caminhando ficando a menos de 500m de sua ex-companheira.

Neste momento o Agente de Recomendação Baseado em Segurança e Vigilância Ubíquas (ARBSVU), componente integrante da arquitetura do UFollower, já estava monitorando o Morador_2 quando ele ficou a menos de 3km da Morador_1. Quando ele identifica que o Morador_2 passa a ficar a menos de 500m Morador_1 ele notifica 3 atores: a retaguarda do modelo – perfil de administrador, o agente de segurança mais perto e a própria Morador_1, representado no modelo pela mulher. As mensagens são personalizadas dependendo do perfil a ser notificado. A identificação da distância entre os dois atores acontece pela captura das coordenadas emitidas pela Tornozeleira (TORNOZELEIRA001) do Morador_2 e das localização geográfica do smartphone da Morador_1.

Embora o cenário de deslocamento está previsto no modelo um histórico de contexto foi criado tendo como parâmetro o veículo de placas IZA-8J32. Esse veículo foi identificado por 3 câmeras (cam_1, cam_2 e cam_3) da simulação e, ao consumir os serviços disponibilizados pela API do UFollower, foi identificado como proprietário do veículo o Morador_2.

A Tabela 8 ilustra parte do histórico desse veículo próximo ao ambiente de trabalho da Morador_1.

Tabela 8: Histórico de Contexto veículo

<i>Dia</i>	<i>Hora</i>	<i>Dia da Semana</i>	<i>Câmera</i>	<i>Endereço</i>	<i>Placa</i>
27/03/2020	09:32	sexta-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
30/03/2020	09:05	segunda-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
31/03/2020	14:12	terça-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
01/04/2020	11:08	quarta-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
02/04/2020	15:15	quinta-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
03/04/2020	11:44	sexta-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
06/04/2020	08:56	segunda-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
07/04/2020	16:00	terça-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
08/04/2020	10:10	quarta-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
09/04/2020	14:07	quinta-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
10/04/2020	11:00	sexta-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
13/04/2020	11:58	segunda-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
14/04/2020	15:49	terça-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
15/04/2020	10:56	quarta-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32
16/04/2020	14:44	quinta-feira	Câmera 5	Rua 24 de Outubro	IZA-8J32
17/04/2020	11:35	sexta-feira	Câmera 4	Av. Plínio Brasil Milano	IZA-8J32

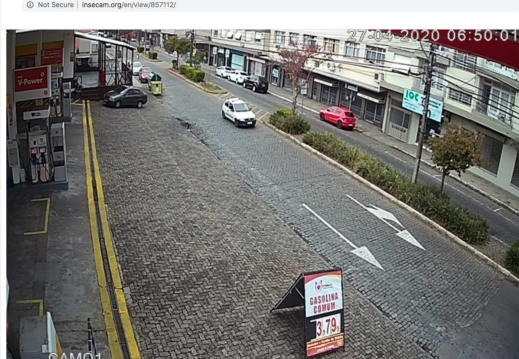
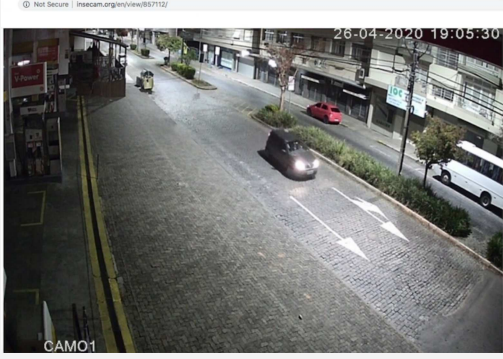

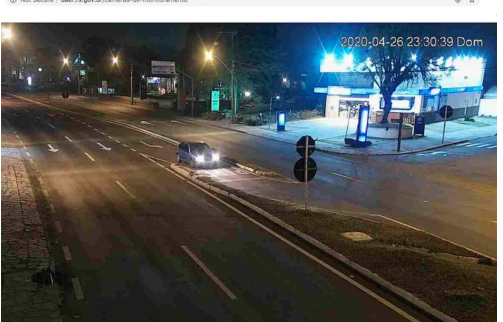
Fonte: Elaborado pelo Autor.

Conforme pode ser verificado na Tabela 8, embora próximo do ambiente de trabalho da Morador_1, até o dia 17/04/2020 não tinha sido emitido alerta em virtude de o veículo não ter infringido do limite mínimo de 500m da Morador_1. No entanto, o modelo já conhecia o padrão do veículo e o padrão de dias de semana e seus horários. As segundas-feiras, terças-feiras e sextas-feiras o veículo sempre foi identificado nas manhãs. Nas terças-feiras e quintas-feiras o veículo foi identificado nas tardes.

A câmera 4 está na avenida Plínio Brasil Milano instalada a 3.7km do local de trabalho da vítima. A câmera 5 está instalada na rua 24 de Outubro, a qual fica a 4.6km do local de trabalho da vítima.

Há diversas câmeras espalhadas por muitas cidades do Rio Grande do Sul, as quais podem ser utilizadas pelo modelo através de convênio ao até mesmo de forma gratuita em função de terem o *link* de acesso aberto. A Tabela 9 apresenta duas dessas câmeras que foram utilizadas na simulação do modelo. A câmera 01 é tem seu *link* fornecido pelo Insecam, um diretório on-line de câmeras IP, nele as câmeras não possuem proteção por senha e transmitem seus *feeds* de vídeo ao vivo com centenas de milhares de visitantes diariamente (XU; CHEN, 2018). A câmera 02 tem como fornecedor de link DAER-RS Departamento Autônomo de Estradas de Rodagem. Trata-se de uma autarquia estadual vinculada à Secretaria de Logística e Transportes que, entre outros, realiza o monitoramento em rodovias através de câmeras (DAER, 2020).

Tabela 9: Câmeras de acesso aberto utilizadas na simulação do modelo

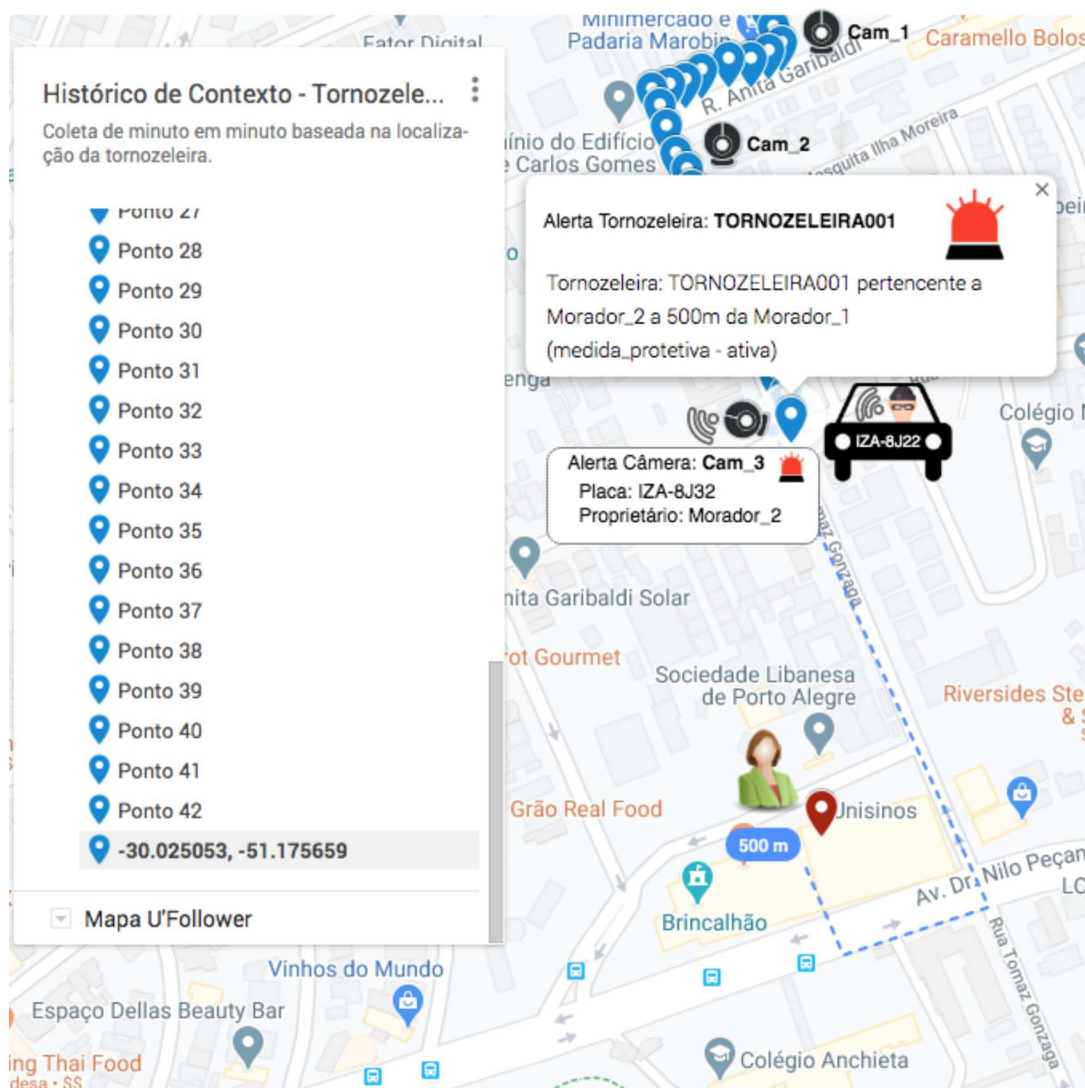
Id	<i>Dia</i>	<i>Noite</i>
01		
02		

Fonte: Elaborado pelo Autor.

A terceira câmera utilizada na simulação do modelo não consta na Tabela 09 em função de ter sido necessário simular um alerta para ela.

Mantendo o padrão do histórico de contexto já conhecido pelo modelo, na segunda-feira 20/04/2020, durante o período da manhã, o mesmo veículo também foi identificado pelas Câmeras 1, 2 e 3, infringindo a medida legal conforme pode ser verificado na Figura 46.

Figura 46: Morador_2 já vinha sendo monitorado antes da emissão do alerta



Fonte: Elaborado pelo Autor.

Na Figura 46 é possível identificar os seguintes cenários:

- 43 pontos de monitoramento para o veículo de placas IZA-8J32 foram criados. Esse acompanhamento é necessário em virtude do modelo realizar de forma dinâmica o cálculo da distância entre o ponto em que o veículo se encontra com o ponto onde a Morador_1 está.
- Quando o veículo de placas IZA-8J32 passou a adentrar um perímetro inferior a 3km, considerando a posição da Morador_1, as câmeras Cam_1 e Cam_2 começaram a monitorá-lo. Nesse momento ainda não existe a necessidade de nenhuma notificação, pois as duas câmeras estão a mais de 500m da Morador_1, distância que não infringe medida legal estabelecida.
- No 43º ponto, de coordenadas $X = -30.025054$ e $Y = -51.175669$, a Cam_3 também identificou o veículo e seu condutor. O primeiro foi identificado por consulta de placa, mesma requisição feita pelas outras câmeras. O segundo, foi

identificado por sinal de GPS da tornozeleira que o condutor traz consigo. Neste ponto o modelo UFollower deixa sua atuação passiva ao contexto e passa a ter um comportamento ativo. Isso se deve ao fato de a Cam_3 estar a 500m da Morador_1, ou seja, passando dela, o veículo fica a menos de 500m, descumprindo medida legal e, portanto, podendo ser interceptado por algum Agente da simulação.

Para as situações representadas na Figura 46 há mensagens configuradas na simulação. As mensagens são personalizadas em virtude do perfil cadastrado no modelo. A Tabela 10 ilustra essas mensagens:

Tabela 10: Mensagens personalizadas no modelo UFollower

<i>Perfil</i>	<i>Mensagem</i>
<i>Retaguarda Web</i>	“Atenção! O nodo pessoal do Morador_2 está infringindo medida legal contra o nodo pessoal da Morador_1. Morador_2 está a 0.5km da Morador_1. Agente_1 deslocado para Rua Afonso Taunay, Boa Vista, Porto Alegre.”
<i>Agente_1</i>	“Morador_2 descumprindo medida legal.”
<i>Morador_1</i>	“Atenção! O nodo pessoal do Morador_2 está perto de você. Não se preocupe, a situação já foi identificada e o Agente_1 já foi deslocado para interceptá-lo.”

Fonte: Elaborado pelo Autor.

Observa-se na aplicação deste cenário que analisando o histórico de um morador com o seu veículo foi possível, pelo dia da semana e horário, inferir que a moradora estaria no seu ambiente de trabalho, local para onde ele estava se dirigindo. A simulação se mostrou eficiente, visto que entre os vários objetos envolvidos na simulação (nodos pessoais, câmeras e veículos) foi corretamente apontando qual veículo foi identificado por quais câmeras, qual o morador o conduzia, o sinal de sua tornozeleira eletrônica e qual morador tinha medida protetiva ativa.

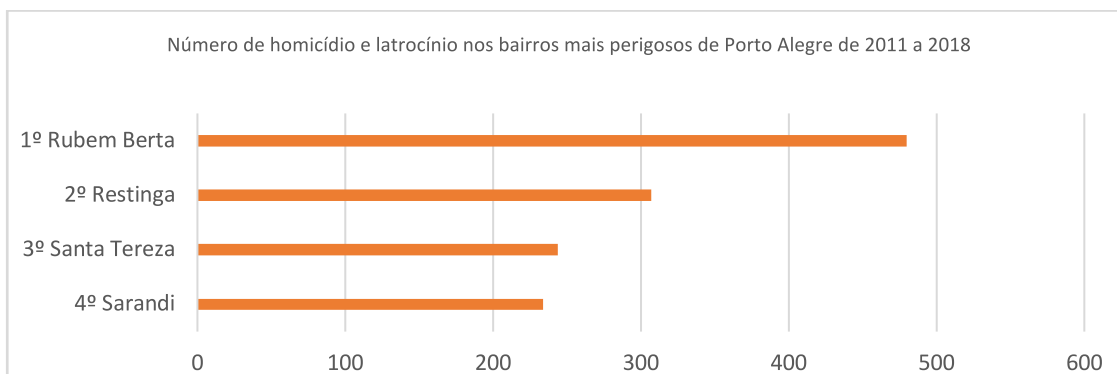
O cenário levou em consideração histórico de contexto, geolocalização e um ambiente de cidade inteligente simulado. Como nenhum dos trabalhos relacionados utilizou histórico de contextos, a contribuição do modelo UFollower nesse cenário foi utilizá-lo para proporcionar o acompanhamento ubíquo, o que torna evidente o seu diferencial.

5.2.3 Cenário 3

Entre os anos de 2011 e 2018, diversos assassinatos aconteceram em Porto Alegre. É possível discriminar essa informação em 3 grandes grupos: os bairros mais violentos, os dias da semana onde há mais incidência desse tipo de crime e qual o horário em que mais acontecem. Para fins de exemplificação, abaixo os gráficos 1, 2 e 3 ilustram os crimes de homicídio e latrocínio na cidade de Porto Alegre.

Na Figura 47 é possível verificar que existe o mapeamento desses crimes e suas quantidades por bairro na capital do Rio Grande do Sul. Pode-se verificar que, segundo o levantamento, os bairros Rubem Berta, Restinga, Santa Tereza e Sarandí representam, respectivamente, o primeiro, o segundo, o terceiro e o quarto bairro mais perigosos de Porto Alegre.

Figura 47: Relação de crimes por bairros em Porto Alegre

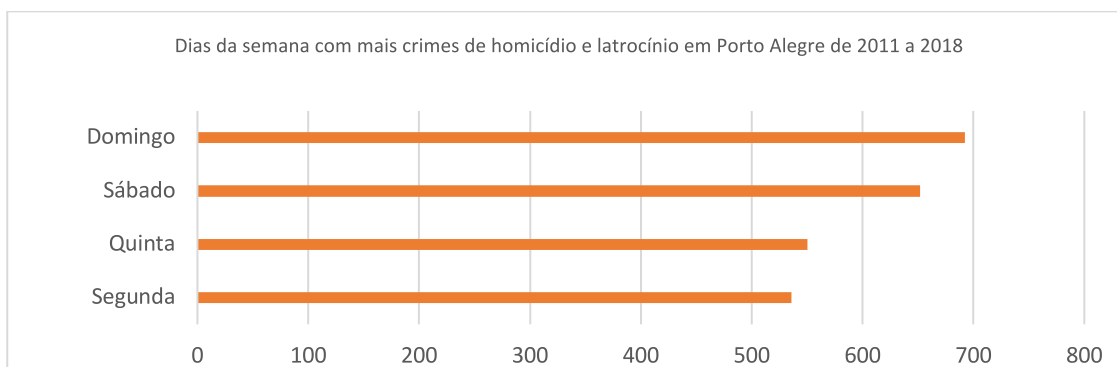


Fonte: TORRES (2018).

Uma vez que o modelo tenha a informação de quais são os bairros mais perigosos de uma determinada região, poderá atuar caso, por exemplo, alguém com o perfil de turista esteja se encaminhando para uma dessas localidades, sem conhecer o perigo do lugar.

Na Figura 48 é possível verificar que há também mapeado para quais dias da semana esses crimes mais acontecem. Nos dias de domingo, sábado, quinta-feira e segunda-feira, respectivamente, acontecem mais os crimes de homicídio e latrocínio. Portanto, segundo com a utilização do exemplo anterior, caso um turista esteja visitando a cidade de Porto Alegre em algum desses dias, o modelo UFollower irá considerar um peso maior na vigilância ubíqua oferecida.

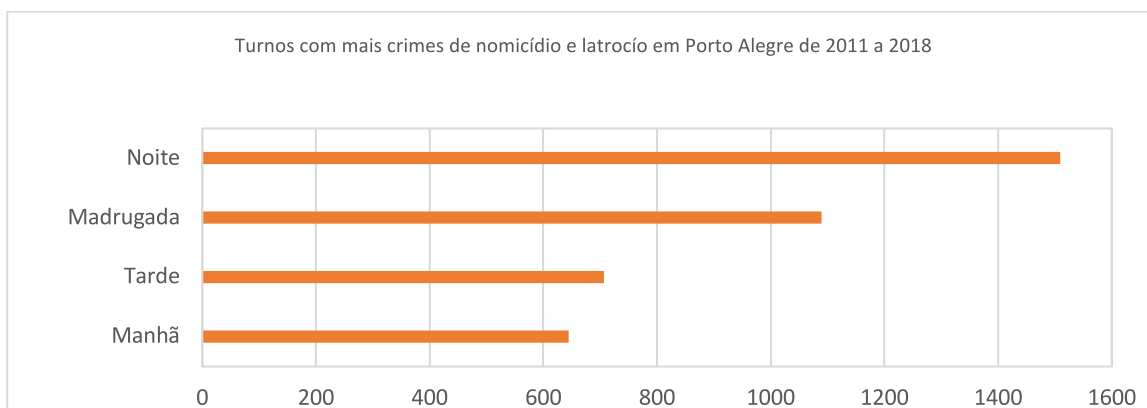
Figura 48: Relação de crimes com os dias da semana



Fonte: TORRES (2018).

A Figura 49 ilustra em qual turno é mais propício ser vítima de um desses crimes. O turno da noite, comparado com os demais turnos apresenta a maior ocorrência dos crimes de homicídio e latrocínio.

Figura 49: Relação crimes com os turnos do dia



Fonte: TORRES (2018).

De forma conclusiva, pode-se inferir que o pior cenário para um turista na cidade de Porto Alegre, considerando-se esses dois crimes hediondos é dirigir-se ao bairro Rubem Berta, em um domingo à noite.

Dadas essas relações apresentadas, é possível introduzir essas informações no modelo para inferir se alguém, fazendo turismo por Porto Alegre e desconhecendo esses dados, terá mais chances de ser vítima ou não de um desses 2 crimes.

Para simular esse cenário foram introduzidos na simulação 3 turistas, cada um em uma localização, conforme mostra a Tabela 11.

Tabela 11: Turistas introduzidos na simulação

<i>Turista</i>	<i>Sexo</i>	<i>Localização</i>
<i>Turista 1</i>	Feminino	(-30.025973, -51.120033) Limite do bairro Passo das Pedras e bairro Rubem Berta.
<i>Turista 2</i>	Masculino	(-30.034505, -51.231191) Bairro Centro Histórico.
<i>Turista 3</i>	Masculino	(-30.134919, -51.231713) Bairro Ipanema.

Fonte: Elaborado pelo Autor.

A Turista_1 é uma turista dos Estados Unidos da América e está no Brasil para fazer turismo. Ela soube da existência do UFollower através de campanhas de divulgação do Estado do Rio Grande do Sul, aceitou os termos de uso do aplicativo e começou a utilizá-lo desde que chegou em Porto Alegre.

Entre os itens coletados do nodo pessoal estão: o dia da semana, geolocalização e o horário, permitindo-se inferir o turno em que ela está. Desta forma, caso a Turista_1, de forma inconsciente, começa a caminhar para uma fronteira de um dos bairros citados na Figura 46, ela será notificada sobre essa situação. Na figura 50 é possível verificar o preenchimento do formulário turista no modelo.

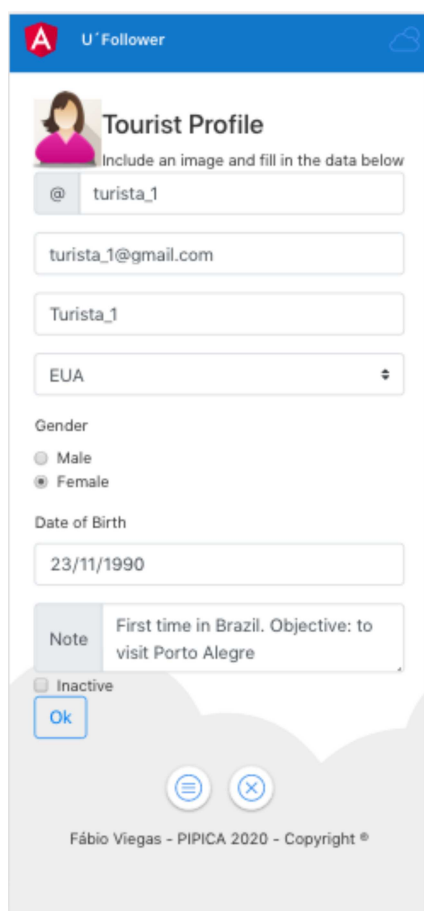
De forma inicial, foi configurada apenas a linguagem inglês para o perfil turista. Portanto, quando este perfil utiliza o aplicativo do UFollower, as interfaces e as notificações serão apresentadas na língua inglesa. Na interface de cadastro, o turista poderá deixar uma nota

registrada. Essa nota poderá ser útil para o módulo de retaguarda, caso seja necessário inferir alguma informação adicional. No exemplo, a turista_1 escreveu que é a primeira vez dela no Brasil e que seu objetivo é visitar a cidade de Porto Alegre. Desta forma, como se trata da primeira vez dela na cidade, fica o alerta para a segurança ubíqua acompanhá-la para evitar que ele entre um território onde já é conhecido o seu índice de determinados tipos de crimes. Uma vez identificado, por geolocalização e histórico de contextos, que a turista está se dirigido para uma região perigosa, será enviada uma notificação para o seu nodo pessoal, informando-a sobre essa situação.

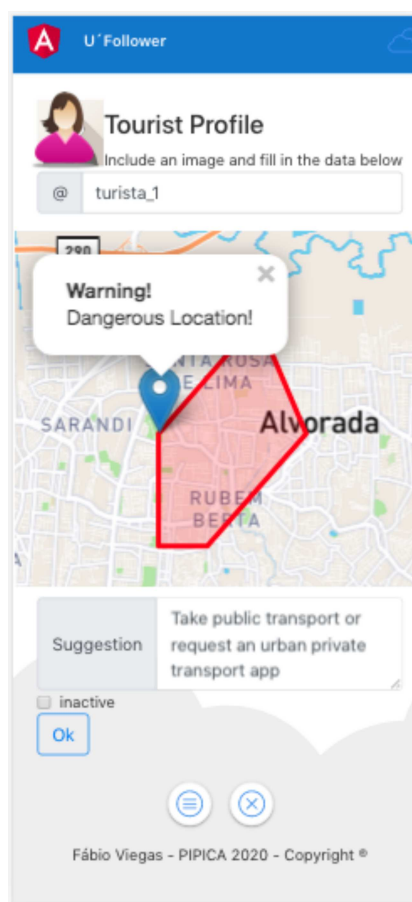
Além da notificação, na mesma interface é trazida ao usuário uma sugestão de como ele pode sair daquela situação. No exemplo citado, onde a turista está quase entrando em uma zona de risco, o UFollowerServer sugeriu que ela pague um transporte público ou que chame ou aplicativo de transporte privado para sair daquela região.

Figura 50: Perfil turista no aplicativo

Figura 51: Turista_1 sendo notificada



Fonte: Elaborado pelo Autor.



Fonte: Elaborado pelo Autor.

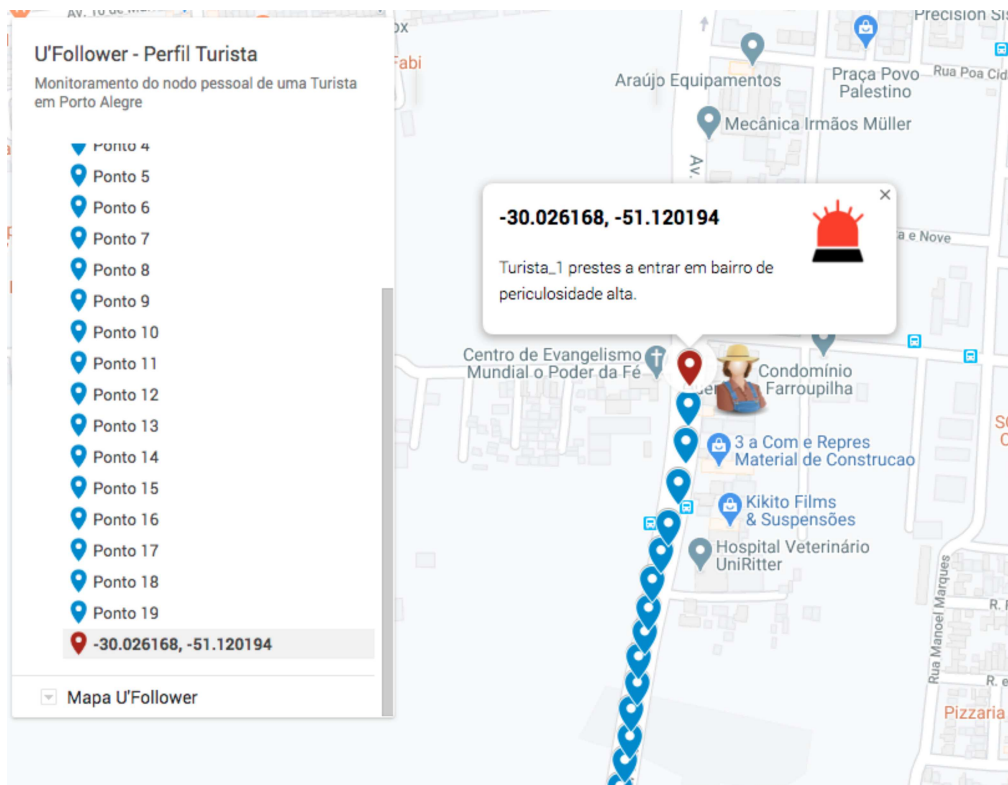
Conforme ilustra a Figura 51, além de notificar a Turista_1 sobre a localidade perigosa para a qual ela está se dirigindo, o UFollower sugere uma ação para que ela possa tomar para sair dessa região.

A simulação nesse caso aconteceu estimando 500m antes da turista ingressar em uma área sabidamente perigosa. Dessa forma, 20 coordenadas geográficas reais de Porto Alegre foram colocadas na simulação. Enquanto o *script* executava, representando o monitoramento a

cada minuto da posição da turista, cada conjunto de coordenada X,Y era verificada, utilizando uma biblioteca geoespacial, onde a coordenada foi aferida contra a área limítrofe da região perigosa.

No vigésimo ponto, foi identificado pelo modelo que o a Turista_1 estaria prestes a ingressar em uma área não adequada, e portanto uma notificação também foi emitida para a retaguarda do UFollower, conforme pode ser verificado no Figura 52.

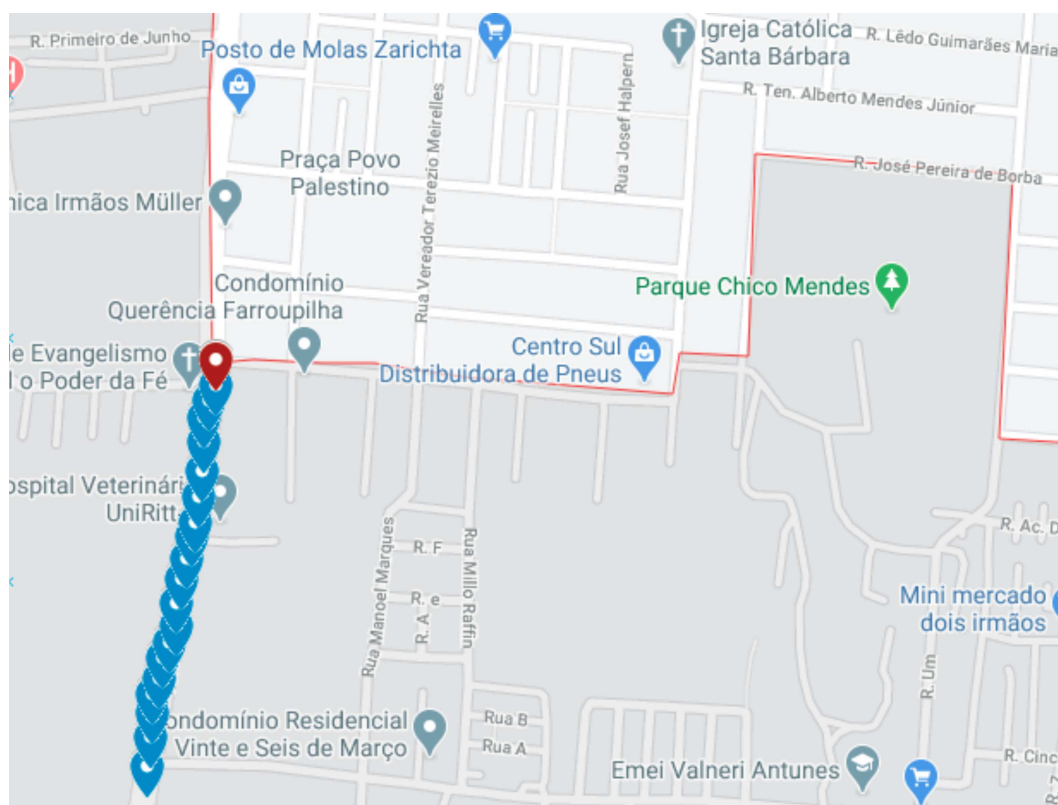
Figura 52: Notificação para a retaguarda do modelo.



Fonte: Elaborado pelo Autor.

Simulando novamente o percurso realizado pela Turista_1, projetando-se as coordenadas em um mapa onde se pode perceber mais claramente onde inicia a área onde há perigo, fica claro que a notificação aconteceu antes dela ingressar na região. Essa simulação pode ser verificada na Figura 53, onde o polígono tracejado em vermelho representa a área de risco, os pontos azuis representam o deslocamento caminhando, bem como o ponto vermelho, onde a notificação foi emitida.

Figura 53: Identificação da área limítrofe da região perigosa



Fonte: Elaborado pelo Autor.

As notificações emitidas e representadas pelas figuras 49 e 50, fazem sentido e mostram-se eficientes se considerado o perfil de quem está sendo monitorado, possibilidade essa permitida pelo UFollower. Por exemplo, caso o perfil seja de um morador, não fará sentido avisar-lhe sobre a periculosidade da região, pois ele já habilita na localidade. Poderá fazer sentido uma outra notificação, como a de um identificador de tiro, os conhecidos *ShotSpotters*. Se um morador está se deslocando para sua residência e um tiro é identificado, pode ser interessante apresentar-lhe essa informação.

Muitas cidades utilizam a tecnologia de detecção de tiro para diferenciar entre tiros e não tiros, em um sistema escalável, de baixo custo e alta precisão, o que aumenta os padrões de segurança para os moradores (MOREHEAD et. Al, 2019). No entanto, por uma questão de limitação de escopo, a simulação com identificadores de tiro não foi implementada na versão inicial do UFollower.

De forma semelhante ao aplicativo no cenário 2, este cenário utilizou de forma inédita, comparando-se com os trabalhos relacionados, a utilização de histórico de contextos, bem como a utilização de geolocalização, ambiente inteligente simulado e de uma biblioteca geoespacial para promover o acompanhamento ubíquo do turista.

6 CONSIDERAÇÕES FINAIS

O trabalho apresentou o modelo UFollower, focado na segurança ubíqua em cidades inteligentes. O objetivo foi simular uma cidade inteligente com suporte a *smart surveillance*. A infraestrutura proposta oferece suporte, uma vez que monitora ambientes, recursos e usuários, integrando-os. O modelo utiliza perfis dos usuários, monitora e gerencia recursos disponibilizando as informações, por meio de recursos de computação vestível, computação em nuvem e IoT, em dispositivos móveis e em *site Web*. A computação ubíqua é o ponto de partida para o modelo proposto, integrando as demais tecnologias, inclusive possibilitando a adição de outros dispositivos, tais como os próprios *wearables* ou outros sensores. Baseando-se nos estudos relacionados, os resultados apresentados neste trabalho podem ajudar a amenizar os problemas encontrados nesta literatura.

6.1 Contribuições

Este trabalho propõe o modelo UFollower o qual contribui com a segurança nas cidades inteligentes, atuando sobre os parâmetros de coleta em vindos ou do meio ou do próprio usuário, baseado em seu perfil. Os históricos de contextos de geolocalização são analisados, permitindo que a coleta se adapte e seja otimizada. A adaptação consiste em efetuar abordagens de Agentes e notificações aos usuários, caso estes estejam em situações de risco.

De acordo com o contexto de onde o usuário se encontra, sensores de contexto (GPS) enviam essas informações para modelo o qual, através de simulação, entrega a melhor alternativa para o momento. Tendo como base a revisão bibliográfica apresentada na seção de trabalhos relacionados, a contribuição científica do UFollower consiste em oferecer uma vigilância ubíqua assistida, otimizada pela avaliação de históricos de contextos.

Para a avaliação do modelo foi abordada a simulação de contextos, tendo em vista os objetos pertinentes à composição de uma cidade inteligente e adotada a vigilância ubíqua. Esses contextos consistem no sinal de GPS emitido por nodos pessoais e tornozeleiras eletrônicas, bem como a avaliação das incidências de violência doméstica analisadas por Redes Bayesianas.

O simulação ainda permite exibir o mapa onde ocorreu uma situação de risco, utilizando coordenadas reais na cidade de Porto Alegre. Dessa forma, tanto o usuário quanto a retaguarda do modelo podem tomar decisões frente ao que lhes é notificado. Essas notificações são configuradas em função do perfil escolhido pelo usuário ao instalar o aplicativo do UFollower.

Apesar da avaliação ter sido usada para situações reais de deslocamento de nodos pessoais, o modelo permite ser usado em diferentes situações onde seja necessário a identificação de situações de risco voltados à segurança pública.

Este tipo de trabalho poderia ser inicialmente comparado com soluções atuais voltadas para a segurança, como a delegacia *online*. Porém, muitas vezes a segurança é tratada de forma reativa, apenas quando o problema já ocorreu é que, se for possível, o usuário procura as autoridades.

Já ao utilizar a solução proposta neste trabalho, tem-se a situação proativa dos agentes de segurança. A possibilidade de um fato acontecer foi prevista pelo modelo, evitando que alguns dos índices apresentados nesse trabalho pudessem ser incrementados. A infraestrutura utilizada na simulação é conhecida e amplamente utilizada, sendo fácil de ser encontrada no dia a dia dos usuários, o que não torna o projeto caro.

O que se pode aproveitar disso tudo é o foco das abordagens ao usuário e na segurança ubíqua preventiva. Talvez ao se utilizar uma solução preventiva, muitos crimes poderão ser evitados com uma intervenção de agendas deslocados pelo modelo. No entanto, quando o problema já existe a melhor solução é utilizar os procedimentos atualmente adotados.

O modelo desenvolvido neste trabalho, em comparação com os trabalhos relacionados avaliados é o único que demonstra a utilização concomitante de histórico de contextos, perfis de usuários e inferência baseada em modelos probabilísticos conhecidos. Desta forma, seu diferencial e contribuição científica ficam baseados nessas utilizações.

Este trabalho também possui limitações e possibilidades de trabalhos futuros. Estes serão abordados no próximo subcapítulo.

6.2 Limitações e Trabalhos Futuros

O modelo possui algumas limitações no que diz respeito ao seu funcionamento. Ele é dependente de sinal GPS, por exemplo, caso o usuário não esteja portando seu *smartphone* ou a tornozeleira de um nodo monitorado deixar de emitir seu sinal, não é possível efetuar as ações propostas no modelo.

Em se tratando de utilização de nodos pessoais (*smartphone*) e dispositivos vestíveis (tornozeleira) a bateria também é um fator limitante. Pois mesmo que o nodo esteja como sinal de GPS ativo, caso a bateria termine, também não será possível efetuar as ações propostas.

Na avaliação feita pelas Redes Bayesianas também podem ter sua limitação devido ao fato de que, em muitos casos, verificou-se que a mulher agredida não tem uma segunda oportunidade para denunciar o agressor.

Futuramente será necessário fazer estudos referentes a escalabilidade do modelo. O que será necessário para possuir diversos usuários utilizando a solução proposta e como isso afetaria a performance das soluções apresentadas (aplicativo, retaguarda *Web*), por exemplo. A coleta de informações de *ShotSpotters* espalhados pela cidade também é uma oportunidade que ficou em aberto.

Também fica como sugestão de melhora a aplicação de redes neurais para classificar imagens para detectar padrões e correlações, análoga à forma como humanos apreendem e relacionam. Para ficar no mesmo padrão e linguagem nos quais o *UFollower* foi concebido, sugere-se a utilização de TensorFlow para essa demanda. Uma vez que se trata de um biblioteca de aprendizado de máquina, de código aberto, feita em também em Python. Com essa demanda desenvolvida, será possível, por exemplo, identificar uma arma na mão de um morador do modelo. Feita essa classificação, poder-se-á verificar se o morador em questão tem uma arma em seu nome, registrada. Se não tiver, uma abordagem de algum agente poderá ser prevista.

Sugere-se também a parametrização de regionalização para o aplicativo, em virtude de apenas ser possível atualmente a configuração de língua inglesa para o perfil turista. Caso esta sugestão seja atendida, o turista que estiver utilizando o aplicativo poderá escolher em qual língua irá querer ser atendido pelo modelo.

REFERÊNCIAS

ABOWD, G. D.; DEY, A. K.; BROWN, P. J.; DAVIES, N.; SMITH, M.; STEGGLES, P. Towards a Better Understanding of Context and Context-Awareness. In: International Symposium On Handheld And Ubiquitous Computing, 1., 1999, London, UK, UK. Proceedings. . . Springer-Verlag, 1999. p. 304 – 307. (HUC 99).

AGARWAL, Devansh, "Cloud Computing- A Brief Introduction", *Medium*, 2019. [Online]. Disponível em: <<https://medium.com/@devanshagarwal121/cloud-computing-a-brief-introduction-393b7928e964>>. Acesso em: 22 Set. 2019.

ALBINO, V.; BERARDI, U.; DANGELICO, R. M. Smart Cities: definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, [S.l.], v. 22, n. 1, p. 3–21, 2015.

AL-MOTWAKEL, O. A. and ZAHARY, A. T., "Ubiquitous Computing and its Security Concerns," *2019 First International Conference of Intelligent Computing and Engineering (ICOICE)*, Hadhramout, Yemen, 2019, pp. 1-16.

ANGULAR, "One Framework Mobile & desktop. [Online]. Disponível em: <<https://angular.io/>>. Acesso em: 29 nov. 2019.

ANTHOPOULOS, L.; FITSILIS, . Evolution roadmaps for smart cities: Determining viable paths. *Proceedings of the European Conference on e-Government, ECEG*, [s. l.], n. June, p. 27–35, 2013.

ARMASH et al. Efficient Way Of Web Development Using Python And Flask. *International Journal of Advanced Research in Computer Science*, [s. l.], v. 6, n. 2, p. 54–57, 2019.

ARMBRUST, et al. A View of Cloud Computing. *Commun. ACM*, New York, NY, USA, v. 53, n. 4, p. 50–58, Apr. 2010.

BARBA et al. Smart city for VANETs using warning messages, traffic statistics and intelligent traffic lights. In: *Intelligent Vehicles Symposium (Iv)*, 2012 IEEE, 2012. Anais. [S.l.: s.n.], 2012. p. 902–907.

BARBOSA et al. Learning in Small and Large Ubiquitous Computing Environments. In: *IEEE/IFIP International Conference On Embedded And Ubiquitous Computing - Volume 01*, 2008., 2008, Washington, DC, USA. Proceedings. . . IEEE Computer Society, 2008. p. 401–407. (EUC '08).

BARBOSA, J. et al. Trailcare: an indoor and outdoor context-aware system to assist wheelchair users. *International Journal of Human-Computer Studies*, [S.l.], v. 116, p. 1-14, 2018.

BATISTA, M.M; Fariniuk, T.M.D; Mello, S.C.B; Smart surveillance em aplicações recentes no Brasil: um estudo de caso nas cidades de Recife e Curitiba. Edição Temática “Cidades inteligentes: planejamento e gestão da sustentabilidade”, *Revista Gestão e Secretariado. Management and Administrative Professional Review*, 2016.

CABALLER et al. elastic cloud computing cluster. *J. Comput. Syst. Sci.*, Orlando, FL, USA, v. 79, n. 8, p. 1341–1351, Dec. 2013.

CARPENTIERI, B. et al. Securing visual search queries in ubiquitous scenarios empowered by smart personal devices. *Information Sciences*, [S.l.], v. 508, p. 393-404, Jan. 2020.

CHEN, G.; KOTZ, D. *A Survey of Context-Aware Mobile Computing Research*. Hanover, NH, USA: Dartmouth College, 2000.

COCCHIA, Annalisa. Smart City lit review. [s.l: s.n.]. Disponível em: <<http://link.springer.com/10.1007/978-3-319-06160-3>>. Acesso em: 29 nov. 2019.

CSC-19, "Porto Alegre é a 8ª cidade mais inteligente do País. [Online]. Disponível em: <<https://www.connectedsmartcities.com.br/2018/09/10/porto-alegre-e-a-8a-cidade-mais-inteligente-do-pais/>>. Acesso em: 10 out. 2019.

DATTA, Parul; SHARMA, Bhisham. A survey on IoT architectures, protocols, security and smart city based applications. 8th International Conference on Computing, Communications and Networking Technologies, ICCCNT 2017, [s. l.], 2017.

DAER, Quem Somos. [s.l: s.n.]. Disponível em: <<https://www.daer.rs.gov.br/quem-somos>>. Acesso em: 27 abr. 2020.

DEITEL, H. M; DEITEL, P. J. *Java, como programar*. 8 ed. São Paulo: Prentice-Hall, 2010.

DESHMUKH, S; MANE, D; RETAWADE, A. Building a single page application Web front-end for e-learning site. Proceedings of the 3rd International Conference on Computing Methodologies and Communication, ICCMC 2019, [s. l.], n. Iccmc, p. 985–987, 2019.

DEY, A. K.; ABOWD, G. D.; SALBER, D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications. *Hum.-Comput. Interact.*, Hillsdale, NJ, USA, v. 16, n. 2, p. 97–166, Dec. 2001.

DEY, A. K. Understanding and Using Context. *Personal Ubiquitous Comput.*, London, UK, UK, Anais. v. 5, n. 1, p. 4–7, Jan. 2001.

FERNANDA, M; FONSECA, S; FIGUEIREDO, R. M. O Femicídio Como Uma Manifestação Das Relações De Poder Entre Os Gêneros A violência contra a mulher não é um fato recente , desde os primórdios da humanidade as mulheres vêm sendo vítimas de agressões , muitas vezes chegando a óbito. <https://periodicos.furg.br/juris/article/viewFile/7680/5330> [s. l.], p. 49–65, 2018.

FERRAZ, F. S.; ANDRÉ, C.; FERRAZ, G. Smart City Security Issues : [s. l.], n. **UCC '14: Proceedings of the 2014 IEEE/ACM 7th International Conference on Utility and Cloud Computing**. October 2016, p. 842–847, 2014.

FERREIRA, J. E. et al. Smart services: A case study on smarter public safety by a mobile app for University of São Paulo. 2017 IEEE SmartWorld Ubiquitous Intelligence and Computing, Advanced and Trusted Computed, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation, SmartWorld/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI 2017 - , [s. l.], p. 1–5, 2018.

FOX, J.; DONNELLAN, A.; DOUMEN, L. The deployment of an IoT network infrastructure, as a localised regional service. 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), [s. l.], p. 319–324, 2019.

FROZZA, A. A. Frozza, R. d. S. Mello and F. d. S. d. Costa, "An Approach for Schema Extraction of JSON and Extended JSON Document Collections," 2018 IEEE International Conference on Information Reuse and Integration (IRI), Salt Lake City, UT, 2018, pp. 356-363.

GANDODHAR , P. S. and Chaware, S. M. Context Aware Computing Systems: A survey, *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on, Palladam, India, 2018, pp. 605-608.

GARRISON, G.; Kim, S.; Wakefield, R. (2012). Success Factors for Deploying Cloud Computing. *Communications of the ACM*. 55. 62-68. 10.1145/2330667.2330685.

GUNDALINI, B. and TOMIZAWA, G. Mecanismo Dsisciplinar de Foucault e o Panóptico de Nentham na Era da Informação. *ANIMA: Revista Eletrônica do Curso de Direito das Faculdades OPET*. Curitiba PR - Brasil. Ano IV, nº 9, jan/jun 2013. ISSN 2175-7119.

HAMPAPUR et al. Smart surveillance: Applications, technologies and implications. Conference: Information, Communications and Signal Processing, 2004.

HU, L. Ma and J. Zhao. DeepGraph: A PyCharm Tool for Visualizing and Understanding Deep Learning Models, *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, Nara, Japan, 2018, pp. 628-632.

ICHIFUJI, Yu et al. A study for understanding of tourist person trip pattern based on log data of Wi-Fi access points. *Proceedings - 2016 IEEE International Conference on Big Data, Big Data 2016*, [s. l.], p. 2167–2174, 2016.

IJAZ, Sidra et al. Smart Cities: A Survey on Security Concerns. *International Journal of Advanced Computer Science and Applications*, [s. l.], v. 7, n. 2, 2016.

JAMEEL, T.; ALI, R.; ALI, S. Security in Modern Smart Cities: An Information Technology Perspective. 2019 2nd International Conference on Communication, Computing and Digital Systems, C-CODE 2019, [s. l.], p. 293–298, 2019.

JHAJHARIA, S.; PAL, S. K.; VERMA, S. Wearable Computing and its Application. *International Journal of Computer Science and Information Technologies*, [s. l.], v. 5, n. 4, p. 5700–5704, 2014.

JOHANNES H. U.; S. Leyk, Y. C.; Duan W.; Knoblock C. A. Extracting human settlement footprint from historical topographic map series using context-based machine learning, *8th International Conference of Pattern Recognition Systems (ICPRS 2017)*, Madrid, 2017, pp. 1-6.

JUNIOR, A. A. et al. A predictive policing application to support patrol planning in smart cities. 2017 International Smart Cities Conference, ISC2 2017, [s. l.], 2017.

KITCHIN, Rob, *The Real-Time City? Big Data and Smart Urbanism* (July 3, 2013). A revised version of this paper, including two new sections, has been published in *GeoJournal* 79(1):1-

14, 2014.

KOLLER, D.; FRIEDMAN, N. Probabilistic graphical models: principles and techniques.

[S.l.]: MIT press, 2009.

KOMNINOS, N. Smart Cities 1 . Smart cities : toward a new paradigm. [s. l.], v. 3, n. June, p. 0–9, 2018. The Sage Encyclopedia of the Internet, 783-789

LADE, Prasanth et al. Detection of changes in human affect dimensions using an Adaptive Temporal Topic model. Proceedings - IEEE International Conference on Multimedia and Expo, [s. l.], p. 1–6, 2013.

LISBOA, VINÍCIUS. Violence atlas reveals homicide rate in Brazil up 4.2%. Disponível em: <<http://agenciabrasil.ebc.com.br/en/direitos-humanos/noticia/2019-06/violence-atlas-reveals-homicide-rate-brazil-42>><<http://agenciabrasil.ebc.com.br/en/direitos-humanos/noticia/2019-06/violence-atlas-reveals-homicide-rate-brazil-42>>. Acesso em: 09 de Set. de 2019.

LIMA, A.; GARRAFEIL, F. Femicídios no Brasil - Mapa mostra taxas de mortes de mulheres em razão do gênero em cada um dos estados.” [Online]. Disponível em:<<https://especiais.g1.globo.com/monitor-da-violencia/2018/feminicidios-no-brasil/>>. Acesso em: 10 mar. 2020.

LÓPEZ et al. Adding sense to the Internet of Things - An architecture framework for Smart Object systems. Personal and Ubiquitous Computing. 16. 291-308 (2012).

MIRAZ, M. H. et al. A review on Internet of Things (IoT), Internet of Everything (IoE) and Internet of Nano Things (IoNT). 2015 Internet Technologies and Applications, ITA 2015 - Proceedings of the 6th International Conference, [s. l.], v. 2020.

MELL, P. and Grance, T. "The NIST Definition of Cloud Computing" [Online]. Disponível em: <<https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf>> . Acesso em: 29 nov. 2019.

MOCKAROO, "Why is realistic data important?" [Online]. Disponível em: <<https://www.mockaroo.com/>>. Acesso em: 16 abr. 2020].

MOREHEAD et al. "Low Cost Gunshot Detection using Deep Learning on the Raspberry Pi," 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 2019, pp. 3038-3044.

NAVARATHNA, P. J.; MALAGI, V. P. Artificial Intelligence in Smart City Analysis. 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), [s. l.], n. Icssit, p. 44–47, 2018.

NIEMALA, E. and JUHANI L. “Survey of requirements and solutions for ubiquitous software.” MUM (2004).

ONISKO, A.; DRUZDZEL, M. J.; AUSTIN, R. M. Application of Bayesian network modeling to pathology informatics. Diagnostic Cytopathology, [s. l.], v. 47, n. 1, p. 41–47, 2019.

PANG, Yawei et al. SPATH: Finding the Safest Walking Path in Smart Cities. IEEE

Transactions on Vehicular Technology, [s. l.], v. 68, n. 7, p. 7071–7079, 2019.

PITOLI, F. ChronicPrediction: um modelo para prognóstico ubíquo de fatores de risco de doenças crônicas não transmissíveis . 2015. Dissertação (Mestrado em Ciência da Computação) — Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2015.

PERBOLI et al. A New Taxonomy of SmartCity Projects. Transportation Research Procedia, [S.l.], v. 3, n. 0, p. 470 – 478, 2014. 17th Meeting of the {EURO} Working Group on Transportation, EWGT2014, 2-4 July 2014,Sevilla, Spain.

PETROLO, R. V. L. and N. Mitton. Towards a smart city based on cloud of things. In Proceedings of the 2014 ACM International Workshop on Wireless and Mobile Technologies for Smart Cities, WiMobCity '14, pages 61–66, New York, NY, USA, 2014. ACM.

PIEKAS, A. A. S. et al. Aspectos legais e percepções sobre as estratégias para cidades inteligentes e criativas: estudo da cidade de Chapecó (SC). urbe. Revista Brasileira de Gestão Urbana, [s. l.], v. 10, n. suppl 1, p. 197–211, 2018.

PIRO, G. et al. Information centric services in Smart Cities. Journal of Systems and Software, [S.l.], v. 88, n. 0, p. 169 – 188, 2014.

QAMAR, T. et al. Smart City Services Ontology (SCSO): Semantic Modeling of Smart City Applications. Proceedings - 2019 7th International Conference on Digital Information Processing and Communications, ICDIPC 2019, [s. l.], p. 52–56, 2019.

RANDELL, Cliff. Wearable Computing : A Review. Expedition, [s. l.], n. Department of Computer Science, University of Bristol, U.K January 2005, 1996.

RAKITA, Bojan et al. Context-Aware Smartphone Applications Based on Generic Context Supervision Services. 2018 26th Telecommunications Forum, TELFOR 2018 - Proceedings, [s. l.], p. 420–425, 2018.

ROUSE, Margaret. Pervasive computing (ubiquitous computing) [Online]. Disponível em: <<https://Internetofthingsagenda.techtarget.com/definition/pervasive-computing-ubiquitous-computing>>. Acesso em: 29 nov. 2019.

ROOPA, Y. M. , Babu M. R and Dhananjaya B. Context-Aware Computing and Big Data Analytics for IoT Applications, 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 2018, pp. 872-876.

SAINATH, R. V. S. N. Y. et al. Sleep Pattern Monitoring and Analysis to Improve the Health and Quality of Life of People. 2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018, [s. l.], p. 1900–1905, 2018.

SHAFER, Steve. Ten Dimensions of Ubiquitous Computing. Managing Interactions in Smart Environments. Managing Interactions in Smart Environments. Springer, London [s. l.], p. 5–16, 2000.

SHAHROUR, Isam; ALILECHE, Lyes; ALFURJANI, Abdurahman. Smart cities: System and tools used for the digital modelling of physical urban systems. 2017 Sensors Networks Smart and Emerging Technologies, SENSET 2017, [s. l.], v. 2017- Janua, p. 1–4, 2017.

SHIN, Dong Hoon et al. Security protective measures for the ubiquitous city integrated operation center. Proceedings - 3rd International Conference on Broadband Communications, Informatics and Biomedical Applications, BroadCom 2008, [s. l.], p. 239–244, 2008.

SCIKIT, Learn "Getting Started" [Online]. Disponível em: <https://scikit-learn.org/stable/getting_started.html> . Acesso em: 10 mar. 2020.

SMAILAGIC, Asim; SIEWIOREK, Daniel P. Wearable computing. Mobile Computing Handbook, [s. l.], v. 6890, n. 6, p. 3–23, 2004.

SANTI, Liliane Nascimento de; NAKANO, Ana Maria Spanó; LETTIERE, Angelina. Percepção de mulheres em situação de violência sobre o suporte e apoio recebido em seu contexto social. Texto & Contexto Enfermagem, v. 19, n. 3, p. 417-424, 2010.

SRIVASTAVA, Priyanshu; KHAN, Rizwan. A Review Paper on Cloud Computing. International Journal of Advanced Research in Computer Science and Software Engineering, [s. l.], v. 8, n. 6, p. 17, 2018.

SSP. Observatório da Violência contra a Mulher da SSP-RS[Online]. Disponível em: <<https://www.ssp.rs.gov.br/observatorio-da-violencia-contra-a-mulher>>. Acesso em: 10 mar. 2020.

STRASSER M. and ALBAYRAK S. Smart City Reference Model : Interconnectivity for On-Demand User to Service Authentication. International Journal of Supply and Operations Management, [s. l.], v. 3, n. 1, p. 1126–1142, 2016.

SWAGGER, "What Is Swagger?" [Online]. Disponível em: <<https://swagger.io/docs/specification/about/>>. Acesso em: 29 nov. 2019.

TAYEBI, A. et al. Development of a Web-Based Simulation Tool to Estimate the Path Loss in Outdoor Environments using OpenStreetMaps [Wireless Corner], in IEEE Antennas and Propagation Magazine, vol. 61, no. 1, pp. 123-129, Feb. 2019.

TELLES, M.; BARBOSA, J.; RIGHI, R.. Um Modelo Computacional para Cidades Inteligentes Assistivas. iSys - Revista Brasileira de Sistemas de Informação, [s. l.], v. 10, n. 1, p. 52–79, 2017.

TORRES, E. Cinco bairros concentram mais de um terço dos assassinatos de Porto Alegre em sete anos [Online]. Disponível em: <<https://gauchazh.clicrbs.com.br/seguranca/noticia/2018/03/cinco-bairros-concentram-mais-de-um-terco-dos-assassinatos-de-porto-alegre-em-sete-anos-cjehe8qep00zx01r4iug2ag1p.html>>. Acesso em: 29 mar. 2020.

TOUSLEY, S.; RHEE, S. Smart and Secure Cities and Communities. Proceedings - 2018 3rd International Workshop on Science of Smart City Operations and Platforms Engineering, in Partnership with Global City Teams Challenge, SCOPE-GCTC 2018, [s. l.], p. 7–11, 2018.

VIANNA, H.; BARBOSA, J.; COSTA, C.; YAMIN, A. C. U'Ductor: a model for supportingubiquitous chronic disease care management. In: SYMPOSIUM ON COMPUTING

SYSTEMS, 2012., 2012, Washington, DC, USA. Proceedings. IEEE Computer Society, 2012. p. 156–162. (WSCAD-SSC '12).

VIER, J.; GLUZ, J.; JAQUES, P. A. Empregando Redes Bayesianas para modelar automaticamente o conhecimento dos alunos em Lógica de Programação. *Revista Brasileira de Informática na Educação*, [s. l.], v. 23, n. 02, p. 45, 2015.

VOIGTMANN, C.; SCHUTTE, C.; WACKER, A.; DAVID, K. A new approach for distributed and collaborative context prediction. In: *Pervasive Computing And Communications Workshops (Percom Workshops)*, 2013 IEEE International Conference On, 2013. Anais. [S.l.: s.n.], 2013. p. 20–24.

WEISER, M. *The Computer for the 21st Century*. New York: Scientific American, 1991.

WILLIAMS, J.; WEATHERBURN D. Can Electronic Monitoring Reduce Reoffending? [s. l.], n. 12122, 2019. Disponível em: <www.iza.org>. Acesso em: 12 mai. 2020.

XIAOMI, Mi Smart Band 4. [s.l: s.n.]. Disponível em: <<https://www.mi.com/br/mi-smart-band-4/>>. Acesso em: 08 mar. 2020.

XU H.; XU F.; CHEN B. Internet Protocol Cameras with No Password Protection: An Empirical Investigation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, [s. l.], v. 10771 LNCS, p. 47–59, 2018.

YANG L.; ELISA N.; ELIOT N. Privacy and Security Aspects of E-Government in Smart Cities. *Smart Cities Cybersecurity and Privacy*, [s. l.], n. May, p. 89–102, 2019.

ZHANG X.; SUN D.; JIANG G. C. Comparative efficacy of nine different dressings in healing diabetic foot ulcer: A Bayesian network analysis. *Journal of Diabetes*, [s. l.], v. 11, n. 6, p. 418–426, 2019.

ZHAO X. et al. Urban infrastructure safety system based on mobile crowdsensing. *International Journal of Disaster Risk Reduction*, [s. l.], v. 27, n. September 2018, p. 427–438, 2018. Disponível em: <<http://dx.doi.org/10.1016/j.ijdr.2017.11.004>>. Acesso em: 12 mai. 2020.

ZHOU P. et al. Online Learning for Context-Aware Multi-User Package Delivery System with Unmanned Vehicles. *IEEE International Conference on Communications*, [s. l.], v. 2019- May, p. 1–7, 2019.

