

# UFollower: A Model for Smart Cities Based on Ubiquitous Security and Surveillance

F. Viegas, J. Barbosa, R. Kunst, and W. Heckler

**Abstract**—Crime is one of the most critical problems in urban centers, especially in large cities. In this sense, technological solutions are needed to provide security to citizens, contributing to the reduction of crime rates. The present work proposes the UFollower model (Ubiquitous Follower) which meets this scenario. The scientific contribution of this work consists of the use of Context Histories and User Profiles for data analysis focused on ubiquitous security. The use of prediction mechanisms and historical contexts allowed to reach up to 70.35% of inference rate for a particular crime (domestic violence). The comparison with the related works indicates that UFollower is the only proposal that presents the issue of public security with context histories and user profile management. The evaluation was conducted through scenarios, allowing to evaluate the related hypotheses. In this sense, a context simulator was built where twenty objects interacted to allow the evolution. Among them, there are people, vehicles, cameras, and wearables.

**Index Terms**— context history, public safety, ubiquitous computing

## I. 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 [1]. Algumas tecnologias podem ser adotadas para agilizar e facilitar as tarefas do cotidiano, entre elas destacam-se: aprendizado de máquina, computação ubíqua, tecnologias sem fio e Internet das Coisas (IoT, do inglês *Internet of Things*), através de sensores, medidores e dispositivos pessoais [2].

Dentre os diversos segmentos que compõem a estrutura de uma cidade, a segurança pública merece atenção especial. No Brasil, estudos realizados no ano de 2019, conforme dados do Atlas de Violência IPEA (2019), a taxa de homicídios 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 [3]. Este trabalho propõe o UFollower, um modelo para minimizar o percentual de violência e crimes através de um gerenciamento de contextos, acompanhando os usuários e trazendo notificações personalizadas ao perfil dos usuários.

F. Viegas, Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, Brasil, fabioviegas.rs@gmail.com

J. L. V. Barbosa, Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, Brasil, jbarbosa@unisinos.br

R. Kunst, Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, Brasil, rafaelkunst@unisinos.br

W. Heckler, Universidade do Vale do Rio dos Sinos (Unisinos), São Leopoldo, RS, Brasil, wesleiheckler@gmail.com

O UFollower prevê também a inferência de um crime de violência doméstica virar fato consumado de feminicídio, através de recursos tecnológicos de aprendizado de máquina, considerando sinais de *Global Positioning System* (GPS) de telefones celulares e tornozeleiras eletrônicas.

A principal contribuição do trabalho consiste na proposição de um modelo que oferece uma vigilância ubíqua assistida, otimizada pela avaliação de históricos de contextos [4]. O modelo diverge dos demais existentes na literatura, pois reúne conceitos de segurança da informação, recomendação, históricos de contextos, computação ubíqua e inferência.

O texto está organizado em cinco seções. Na segunda seção são apresentados os trabalhos relacionados, descrevendo suas características e comparando-os com o sistema proposto. O UFollower é descrito na terceira seção, apresentando suas principais funcionalidades e detalhando sua arquitetura. As decisões tomadas, as tecnologias utilizadas na implementação do protótipo e avaliação são apresentadas na quarta seção. Por fim, a última seção apresenta as conclusões e trabalhos futuros.

## II. TRABALHOS RELACIONADOS

Os critérios para a escolha dos trabalhos foram baseados em termos sinônimos ao assunto contemplado pelo UFollower e suas derivações. A seguinte *string* de busca foi utilizada nas pesquisas:

“*ubiquitous security*” AND “*assistive environments*”) AND (“*context aware*” OR “*location based services*”) AND (“*Smart Cities*” AND “*Surveillance System*”) OR (“*Smart Surveillance*” AND “*Smart City*”).

Pang et al. [5] apresentaram o aplicativo SPATH (*Safest PATH*, ou seja, o Caminho mais Seguro) que fornece segurança aos cidadãos estadunidenses, garantindo uma navegação segura, caminhando, em cidades inteligentes.

Ferreira et al. [6] propõem um modelo de solução móvel para o Campus Universidade de São Paulo (USP), em virtude dele 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 em relação à segurança e bem-estar da comunidade universitária.

Zhao et al. [7] usaram o conceito de *crowdsensing*, que consiste em uma técnica em que um grande grupo de indivíduos com dispositivos móveis compartilha coletivamente dados e extrai 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. Experimentos de verificação e

avaliação foram realizados na Universidade de Tecnologia de Dalian, incluindo monitoramento de deslocamento, medições de dados e de pontos e coletas.

O trabalho de Batista et al. [8] trouxe o conceito de vigilância ubíqua, contextualizado a partir da noção de cidades inteligentes. A prática de coletar informações do meio evoluiu ao longo do tempo e o advento tecnológico dos últimos anos gerou mudanças substanciais nesse processo. A utilização de tecnologias cada vez menores e mais sofisticadas para fins de vigilância, prevenção e militarização do espaço urbano é parte desse processo e constitui a chamada vigilância ubíqua. A pesquisa identificou de que modo o aparato tecnológico para vigilância instalado vem sendo utilizado nos Centros Integrados de Comando e Controle das cidades de Recife e Curitiba, no sentido de aproveitar o legado dos eventos esportivos.

Babanne et al. [9] apresentaram a vídeo-vigilância e o monitoramento por câmeras como fontes ricas de segurança e investigação. A proposta visa agregar novos elementos que colaborem com o processo de reconhecimento de eventos e geração de alertas, tais como detecção de incêndio, reconhecimento de atividades anormais em áreas sociais, sistema de estacionamento inteligente, entre outros. Os autores enfatizam que, em cidades modernas, a vídeo-vigilância se tornou parte importante para proteção e segurança, fornecendo análise inteligente de vídeo.

Para Muchtar et al. [10], o sistema de vigilância ubíqua é um dos mais importantes serviços fornecidos em uma cidade inteligente. Os autores propuseram uma nova técnica unificada para detectar um objeto em movimento nos vídeos de vigilância. O modelo foi baseado em um classificador de aprendizado profundo. Os experimentos foram realizados com dois conjuntos de dados públicos, contendo imagens de clima ruim, de baixa qualidade e de atividades, tais como pessoas andando a cavalo, caminhando, dirigindo, entre outros.

Khudhair et al. [11] propõem um sistema ubíquo de vigilância para reduzir e impedir roubos, usando Redes Neurais Convolucionais, IoT e nuvem. Para os autores, a vigilância ubíqua desempenha um papel importante em nossos tempos, ajudando na redução da taxa de criminalidade e no monitoramento da situação das instalações. O sistema é composto por nós, onde cada nó é composto por um microprocessador (Raspberry Pi) e uma câmera. O sistema pode ser utilizado tanto em escritórios quanto em residências.

Para fins de comparação entre as propostas, este artigo considerou cinco critérios para a avaliação dos trabalhos. O critério “Segurança” indica se a proposta aborda alguma técnica relacionada à segurança da informação. Para o critério “Recomendação”, considerou-se a geração de notificações. O critério “Histórico de Contextos” avaliou se os trabalhos consideraram o registro de contextos para alguma entrega de informação. O critério “Aspectos de Segurança” avaliou a presença da computação ubíqua em cidades inteligentes, fornecendo recursos de suporte para os usuários através da utilização de dispositivos e sensores. Por fim, o critério “Inferência” avaliou se os trabalhos implementaram algum aspecto de aprendizado de máquina.

Assim, na Tabela I é apresentado o relacionamento dos cinco

critérios de avaliação com os sete trabalhos avaliados. Na avaliação, foi colocada a resposta “Não” caso o trabalho em questão não tenha apresentado evidência do critério avaliado, e “Sim” no caso de o trabalho tê-lo apresentado.

TABELA I  
COMPARATIVO ENTRE OS TRABALHOS RELACIONADOS

Critério	[5]	[6]	[7]	[8]	[9]	[10]	[11]
Segurança	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Recomendação	Sim	Sim	Sim	Não	Não	Não	Não
Histórico de Contextos	Não	Não	Não	Não	Não	Não	Não
Aspectos de Segurança	Sim	Sim	Sim	Sim	Sim	Sim	Sim
Inferência	Não	Não	Não	Não	Não	Sim	Sim

Com relação ao critério segurança, todos os trabalhos oferecem suporte. Em relação a segurança pública, o trabalho Campus USP [6] é mais restritivo do que o trabalho vigilância ubíqua [8] devido à sua área de abrangência ser mais reduzida. Com relação ao critério recomendação, apenas os trabalhos SPATH [5], Campus USP [6] e *Urban Safety* [7] atenderam. Todos os trabalhos apresentaram algum aspecto de segurança. O critério inferência foi atendido apenas pelos trabalhos de Muchtar et al. [10] e Khudhair et al. [11].

Diferentemente dos trabalhos relacionados apresentados na Tabela I, o UFollower atende a todos os requisitos, podendo ser aplicado em larga escala e 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.

### III. MODELO UFOLLOWER

Esta seção apresenta o modelo UFollower, abordando inicialmente como os elementos do modelo são organizados e como eles interagem (modelo conceitual). Logo após, o método que implementa o modelo conceitual é descrito. Por fim, a seção aborda as simulações que utilizam cenários de aplicação do modelo.

#### A. UFollower Modelo Conceitual

No modelo UFollower, os nodos são as entidades responsáveis por compartilhar recursos e informações de contexto, tais como: câmeras, telefones celulares, tornozeleiras eletrônicas e perfis de usuário (agente, morador e turista). Contexto é qualquer atributo que possa identificar a situação de uma entidade [12], por exemplo, a localização geográfica de uma tornozeleira eletrônica.

O cliente móvel interage com o modelo através de seu telefone celular. O cliente *web* é uma retaguarda, onde os dados vindos do cliente móvel podem ser analisados. No UFollowerNode, os agentes de segurança, representados pelo Estado, interagem com o modelo, principalmente recebendo dados para inferência.

A Figura 1 mostra que existem quatro atores principais no modelo: Cliente Móvel, Cliente *Web*, Estado (representando um

agente de segurança) e Dados Externos. O Nodo Pessoal refere-se ao assistente pessoal que um ator Cliente Móvel tenha acesso. 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.

Há recursos externos que não estão diretamente contemplados no modelo, mas que podem ser utilizados, tais como: câmeras em via pública, identificadores de tiro, cercamento eletrônico de uma área geográfica e aplicativos já desenvolvidos e em execução. Esses recursos podem ser utilizados para identificação da placa de um veículo furtado, por exemplo. Dessa forma, o modelo foi projetado para que quaisquer dados externos sejam conectados diretamente com a parte servidora e tenham sua gestão feita pelo Agente de Recomendação Baseado em Segurança e Vigilância Ubíquas. A comunicação entre um nodo pessoal e o servidor é efetuada através de uma *Application Programming Interface* (API) com base em *Representational State Transfer* (REST), onde trafega *JavaScript Object Notation* (JSON).

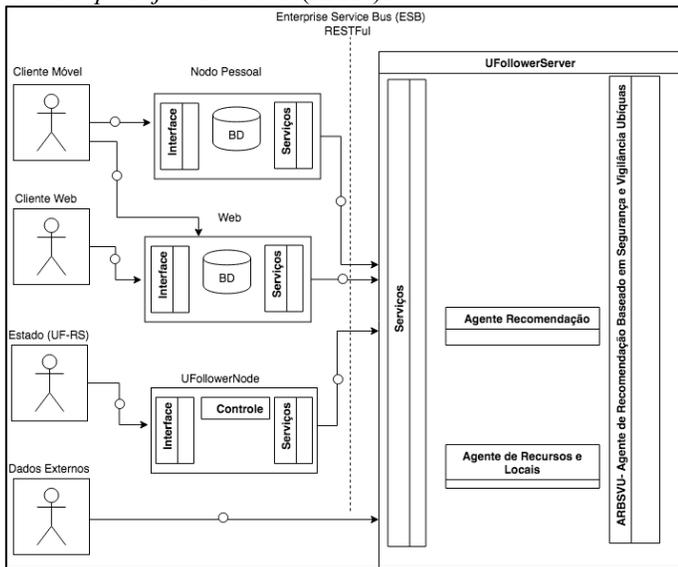


Fig. 1. Arquitetura conceitual do UFollower.

A Figura 2 apresenta a ontologia de Condições de Segurança, que tem por objetivo gerenciar perfil e realizar recomendação, tendo por base o tipo de pessoa, o tipo de ambiente, o tipo de segurança, o tipo de recurso e saúde. 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*. O *TipoSegurança* ativado para um *TipoPessoa* poderá ser “pública”, “privada” ou “crowdsensing”, onde um grande grupo de indivíduos, que possui um *TipoRecurso* capaz de detectar e computar, compartilha dados coletivamente e extrai informações para medir, mapear, analisar, estimar ou inferir qualquer processo de segurança de interesse comum.

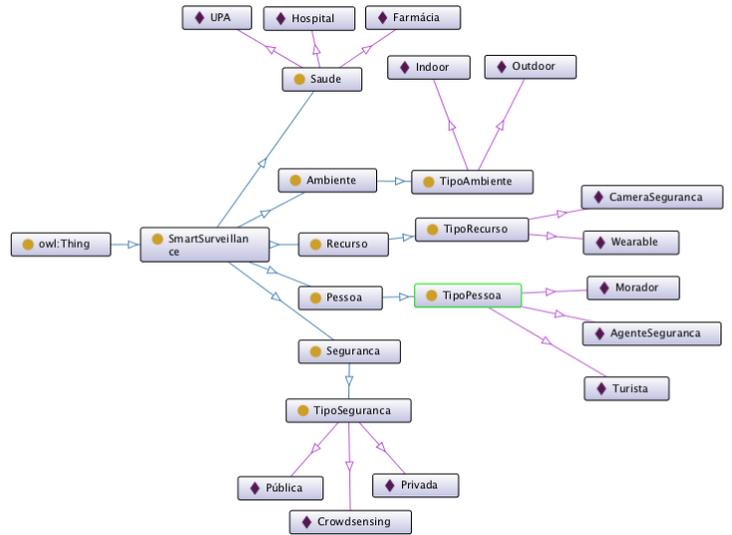


Fig. 2. Representação da ontologia de Condições de Segurança.

A ontologia permite um mapeamento das relações existentes entre os objetos que configuram a vigilância ubíqua e entre cada uma das características que influenciam nos fatores de segurança para que se possa monitorar e, posteriormente, criar uma Rede Bayesiana utilizada no processo de monitoramento da condição de violência doméstica. Redes Bayesianas são modelos probabilísticos que lidam com a representação de conhecimentos em domínios onde existe incerteza [13].

O componente Agente de Recomendação Baseado em Segurança e Vigilância Ubíquas do modelo UFollower contém o SMA (do inglês *System Management Agent*) responsável pelas ações referentes à solicitação de *login*, atualização de dados na plataforma, monitoramento ubíquo, consulta de recursos locais, entre outros. A modelagem deste SMA é apresentada na Figura 3, usando a notação Prometheus [14].

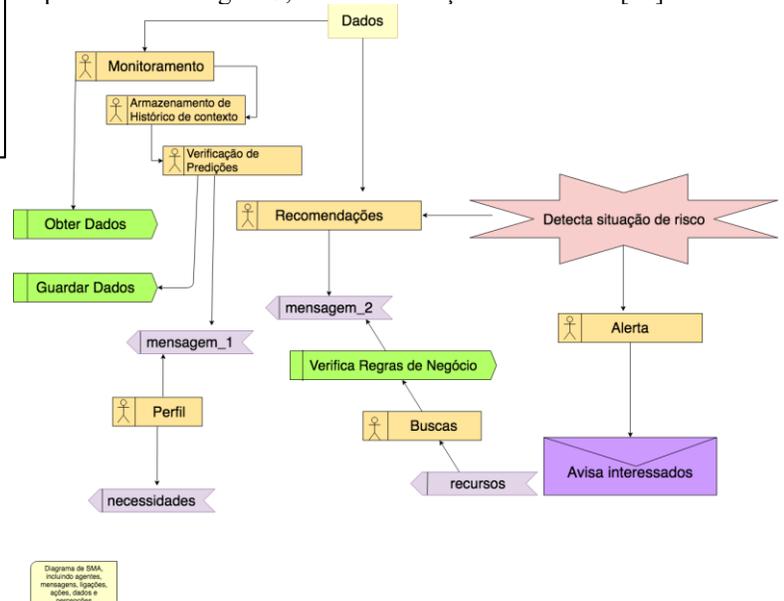


Fig. 3. Sistema Multiagente do modelo UFollower.

O agente *Perfil* tem por objetivo tratar das particularidades de cada usuário, considerando suas preferências e seus contextos. O agente *Monitoramento* recebe e armazena as

informações. O agente *Recomendações* é responsável por gerar as recomendações do modelo. O agente *Buscas* é responsável por descobrir recursos e buscar dados. O agente *Alerta* notifica interessados em receber informações do modelo. O agente de *Armazenamento de Histórico de Contexto* atua guardando determinadas situações. O agente *Verificação de Predições* tem a função de prever determinadas situações com o intuito de auxiliar os usuários em determinados cenários.

### B. UFollower Arquitetura Móvel e Web

Partes fundamentais do modelo, o aplicativo e a retaguarda *web* estão diretamente relacionados às funcionalidades básicas concretizadas. A Figura 4 mostra a arquitetura do aplicativo móvel e do *site web*. Em ambas as arquiteturas há uma representação de um banco de dados local e comunicação entre as camadas, utilizando WiFi na aplicação cliente e *Hyper Text Transfer Protocol Secure* (HTTPS) na aplicação servidor.

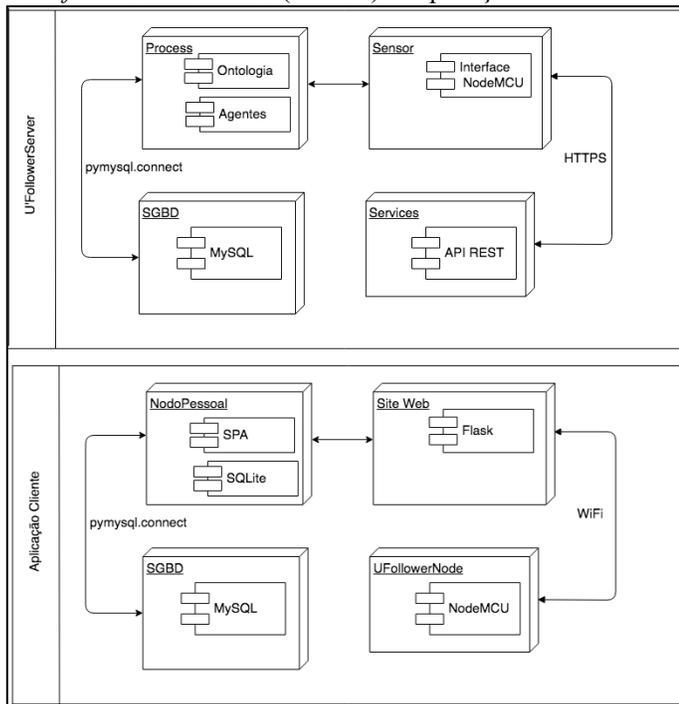


Fig. 4. Arquitetura do aplicativo e do *site web*.

A aplicação servidor (*back-end*) é responsável pelo repositório de dados do estudo de caso e por atender as requisições feitas pela Aplicação Cliente. O módulo de Sensor no Servidor é responsável pelo processamento dos sinais enviados da aplicação cliente responsável pela coleta de dados nos sensores. A Aplicação Cliente (*front-end*) é responsável por interagir com o usuário e por enviar requisições para a Aplicação Servidor. A Nuvem (*web*) é responsável por disponibilizar um serviço para fazer a interligação entre a Aplicação Cliente e Aplicação Servidor.

## IV. IMPLEMENTAÇÃO E AVALIAÇÃO

A definição das tecnologias usadas na implementação foi baseada na experiência dos autores, sendo que as mesmas foram aplicadas com sucesso em outros protótipos.

### A. Aspectos de Implementação

O Python® foi utilizado na implementação do protótipo da

aplicação servidor, apresentado na Figura 4. O servidor *web* foi implementado de forma nativa em Python® através do módulo *http.server*. Esse módulo define classes para implementar servidores *Hyper Text Transfer Protocol* (HTTP) e, embora não seja recomendado para um ambiente de produção por apenas fazer verificações básicas de segurança, para este estudo poderá ser utilizado. Foi utilizado também o banco de dados MySQL®. Python® e MySQL® foram selecionados em função de serem robustos, gratuitos e de domínio de um dos autores deste estudo.

O protótipo da aplicação cliente para a parte do *NodoPessoal* foi desenvolvido no formato de 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 [15]. Na parte *web*, foi desenvolvido um site utilizando Flask, que é um *micro framework* em Python® que fornece a funcionalidade básica da estrutura da *web* e permite mais *plugins* serem adicionados para estender a funcionalidade e o conjunto de recursos [16].

A comunicação entre as aplicações, tanto Cliente (*front-end*) como Servidor (*back-end*), terá como estilo arquitetônico o 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 *Uniform Resource Locator* (URL) único. A escolha por este tipo de serviço *web* é por não estar limitado a retornar dados num determinado formato de arquivo [17].

### B. Aspectos de Avaliação

O modelo foi avaliado através de um estudo de caso, considerando três cenários. Segundo Deshmukh [15], cenários são exemplos de sessões de interação e consistem em descrições de ações sequenciais relacionadas a exemplos da vida real, em vez de descrições abstratas das funções. Nesse sentido, as próximas seções apresentam os cenários de avaliação.

#### 1) Cenário 1 – Inferência de Femicídio com Redes Bayesianas

Com a utilização de 2000 registros de dados realísticos sobre ocorrências de agressão doméstica gerados a partir da ferramenta Mockaroo®, foram aplicadas Redes Bayesianas para a inferência do fato em questão. Essa é uma ferramenta comercial de fácil uso para geração realista de dados, que não requer programação [18]. 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 solicitou 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: *consumado*, *primeira\_agressao*, *medida\_protetiva* e *nova\_denuncia*. O valor 0 em cada coluna indica a não ocorrência da situação. Em contrapartida, o valor 1 indica a ocorrência. Ou seja, no caso do atributo *primeira\_agressao*, o valor 1 indica que o registro se trata da primeira agressão da vítima. Dadas as 3 últimas colunas, procurou-se prever a taxa de acerto da primeira coluna, conforme a amostra de dados apresentada na Tabela II.

TABELA II  
AMOSTRA DE DADOS TABELADOS PARA O MODELO

Consumado	Primeira Agressão	Medida Protetiva	Nova Denúncia
1	0	1	0
1	0	1	1
0	1	0	0
0	1	0	0
1	0	1	1

A implementação usou os seguintes algoritmos baseados em Naive Bayes, utilizando Scikit-learn: MultinomialNB, GaussianNB, ComplementNB, BernoulliNB e CategoricalNB. O algoritmo com a maior taxa de acerto, atingindo o percentual de 70,35%, foi o MultinomialNB.

Dos 2000 registros, 80% foram utilizados para o treinamento do modelo e 20% foram utilizados para os testes. Para validar os algoritmos, foi desenvolvido um algoritmo base que sempre infere o valor 1 para a coluna *consumado*, indicando que o fato foi consumado, independente dos valores das colunas *primeira\_agressao*, *medida\_protetiva* e *nova\_denuncia*. Nessa situação, o algoritmo base acertou 66,17% das vezes. Essa abordagem foi realizada para fins de comparação dos resultados sobre as inferências do UFollower. Entende-se que, caso em um domínio real uma mulher vá a uma delegacia de polícia e, com as informações fornecidas, o modelo infira esse percentual, poderá ser dada uma atenção diferenciada ao caso.

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 conjunto de ferramentas conveniente para fins estatísticos e comportamentais [19].

## 2) Cenário 2 – Prevendo aproximação de um agressor

O segundo cenário usou 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. Na Figura 5 é apresentada a interface principal do aplicativo do UFollower. O aplicativo foi elaborado com a possibilidade de se ter três perfis de usuário e três menus que podem ser utilizados por quaisquer dos perfis.

É possível utilizar o aplicativo com os perfis Morador, Agente e Turista. Cada perfil tem um conjunto de configurações particulares que possibilitam o UFollower entregar informações relevantes a cada perfil. Qualquer perfil pode pedir ajuda, abrir as notificações que já tenham sido enviadas para ele e, ainda, verificar o destino. Essa última possibilidade do menu é fundamental para qualquer perfil. Seja para um agente de segurança solicitar informações sobre o local para onde irá se deslocar, seja para um turista, de maneira preventiva, requerer informações sobre o local para onde deseja ir.

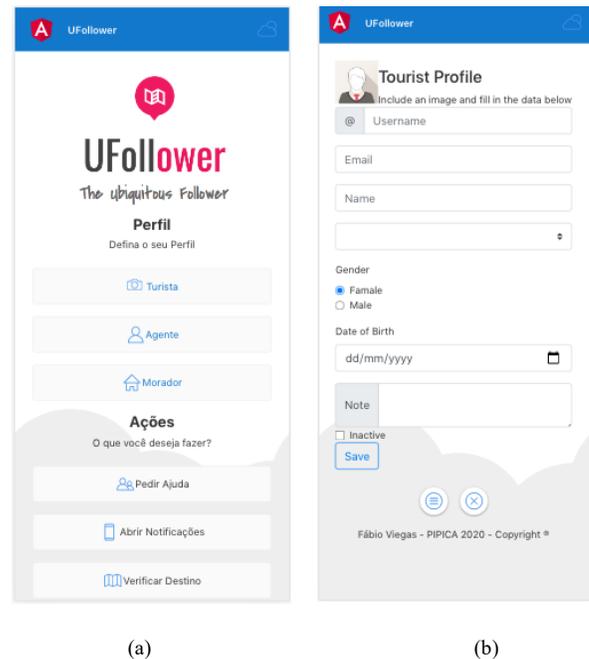


Fig. 5. Interface inicial do aplicativo (a) e interface para o perfil turista (b).

Dessa forma, com a utilização do aplicativo, caracterizado no modelo como a interface direta com o usuário, o modelo é capaz de configurar dados iniciais que serão utilizados para auxiliar no monitoramento. Parte fundamental desse monitoramento é a inferência. Caso um agressor esteja se aproximando de alguém para o qual o UFollower já tenha configurado que não pode se aproximar, então algumas ações são tomadas pelo modelo, que notifica as partes interessadas. Para este cenário foi simulada uma agressão provinda de violência doméstica, onde o objeto vestível (tornozadeira eletrônica) que está de posse do agressor é introduzido na simulação. As Figuras 5(a) e 5(b) identificam, respectivamente, a interface inicial do aplicativo, onde é possível escolher o tipo de perfil, e o cadastro de um perfil turista.

Para o UFollower, a mulher deste cenário tem 34 anos, é brasileira e já foi agredida pelo ex-companheiro, o qual está solto sendo monitorado por tornozadeira. Utilizando georreferenciamento é possível saber a localização tanto da Moradora em questão – através de seu telefone celular, como do monitorado – utilizando a sua tornozadeira.

Um outro perfil do modelo também é acionado nesse cenário: o perfil Agente. O UFollower buscará qual o agente de segurança mais próximo, tanto da Moradora como do Monitorado e enviará para o agente uma notificação em seu nodo pessoal, com a posição da possível vítima e de quem esteja infringindo alguma medida legal. De posse dessa informação, o agente poderá tomar algumas decisões, tais como: se deslocar até o local, solicitar algum reforço para a abordagem ou, se já estiver em algum atendimento, solicitar ao modelo que despache outro colega seu para atender à situação.

No exemplo, o Agente de Recomendação Baseado em Segurança e Vigilância Ubíquas, componente integrante da arquitetura do UFollower, já estava monitorando o *Morador\_2* quando ele ficou a menos de 3km do *Morador\_1*.

Quando ele identifica que o *Morador\_2* passa a ficar a menos de 500m do *Morador\_1* ele notifica 3 atores: a retaguarda do

modelo – perfil de administrador, o agente de segurança mais perto e o próprio *Morador\_1*, representado na Figura 6 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 da localização geográfica do telefone celular do *Morador\_1*.

O objetivo da simulação foi verificar se o proprietário do veículo ao qual a placa está vinculada iria descumprir medida protetiva, aproximando-se demais do local de trabalho de sua ex-companheira. O modelo, conhecendo todos esses artefatos, (quem é a ex-companheira, onde ela trabalha e mora, o ex-companheiro, seu veículo, a medida protetiva dela contra ele) foi capaz de inferir se ele estava se dirigindo ao local de trabalho dela. Para poder simular o comportamento dos objetos foi construído um simulador de contexto em Python®.

A Figura 6 ilustra os vinte objetos inseridos no simulador. Sendo eles doze perfis, três veículos e cinco câmeras. Entre os perfis, havia seis moradores, três turistas e três agentes de segurança. A simulação colocou os vinte objetos a interagirem entre si enquanto ficou-se monitorando a saída da simulação. Na base de dados utilizada, foram definidos os relacionamentos entre os objetos. Havia perfil do tipo morador com medida protetiva contra um outro perfil do mesmo tipo. Ainda, um perfil do tipo morador possuía uma tornozeleira eletrônica, caracterizada como o objeto vestível da simulação. Um contexto de geolocalização foi sendo montado conforme a simulação acontecia. O UFollower conhecia a posição dos nodos pessoais de todos os perfis, bem como das câmeras espalhadas no meio. Desta forma, foi possível extrair inferências. A Tabela III traz uma amostra do histórico de contexto do veículo de placas IZA-8J32, que foi monitorado por um mês no simulador.

TABELA III  
AMOSTRAGEM DO HISTÓRICO DE CONTEXTO DA PLACA IZA-8J32

Dia	Hora	Dia	Câmera	Endereço
27/03	09:32	Sexta	Câm 4	Av. Plínio Brasil Milano
30/03	09:05	Segunda	Câm 4	Rua 24 de Outubro
01/04	11:08	Quarta	Câm 4	Av. Plínio Brasil Milano
07/04	16:00	Terça	Câm 5	Rua 24 de Outubro
09/04	14:07	Quinta	Câm 5	Rua 24 de Outubro

Para o UFollower as câmeras também são parte do contexto, com endereço conhecido. A partir da verificação dos artefatos conhecidos, o modelo notificou o UFollowerServer do descumprimento de medida legal. A Figura 6 ilustra o ecossistema do UFollower em funcionamento. Câmeras, tornozeleira e telefone celular atuando em conjunto. Quando um cenário como o apresentado pela Figura 6 se concretiza, diversas partes interessadas são notificadas pelo modelo. A retaguarda *web* recebe uma notificação para que seja possível despachar uma viatura, caracterizada pelo agente no UFollower, o mais rápido possível, para interceptar o *Morador\_2* antes dele ter acesso ao local de trabalho do *Morador\_1*. A Figura 7 ilustra essa abordagem. Os nodos pessoais de todos os envolvidos também recebem notificações. Inclusive da parte a qual está quebrando o que lhe foi imputado.

Essa notificação serve como uma abordagem preventiva do modelo.

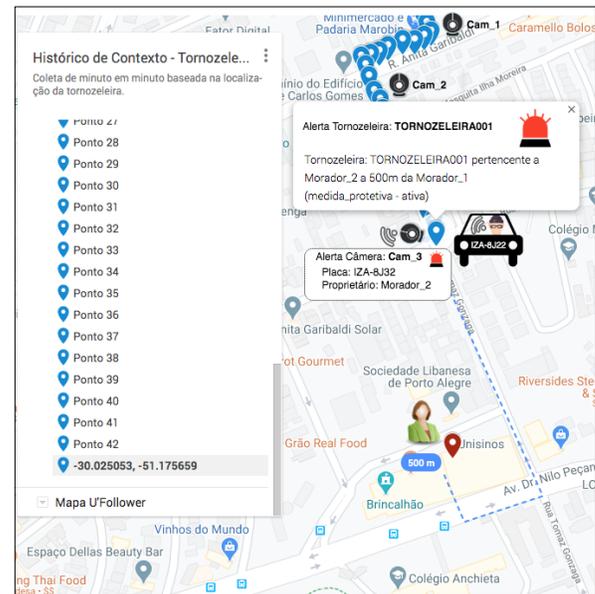


Fig. 6. Perfil Agente do aplicativo recebendo notificação.

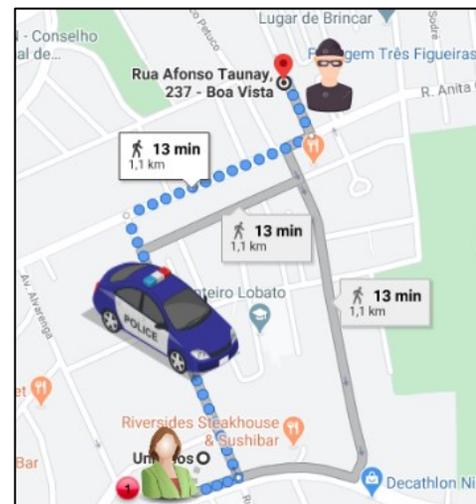


Fig. 7. Agente de Segurança interceptará *Morador\_1*.

### 3) Cenário 3 - Turista sendo monitorado pelo modelo

A *Turista\_1* está no Brasil e 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. Torres [20] relata que, conforme um levantamento da quantidade absoluta de vítimas realizado por dois jornais do Rio Grande do Sul, Rubem Berta, Restinga, Santa Tereza e Sarandi são 4 bairros de Porto Alegre considerados perigosos em relação aos crimes de homicídio e latrocínio.

A Figura 8 ilustra o modelo notificando uma turista através do seu telefone celular. O modelo inferiu que a turista estava prestes a adentrar uma região perigosa, segundo Torres [20], e notificou o UFollowerServer. Também é possível verificar que a inferência aconteceu no vigésimo minuto de monitoramento. Conforme a turista foi se deslocando pela região monitorada, foi

possível deduzir que ela iria para uma zona, a qual o modelo de posse dos dados-crime do lugar, já conhece o perigo. Desta forma, tanto para a retaguarda *web* como para o nodo pessoal da turista, foram emitidas notificações do potencial risco.

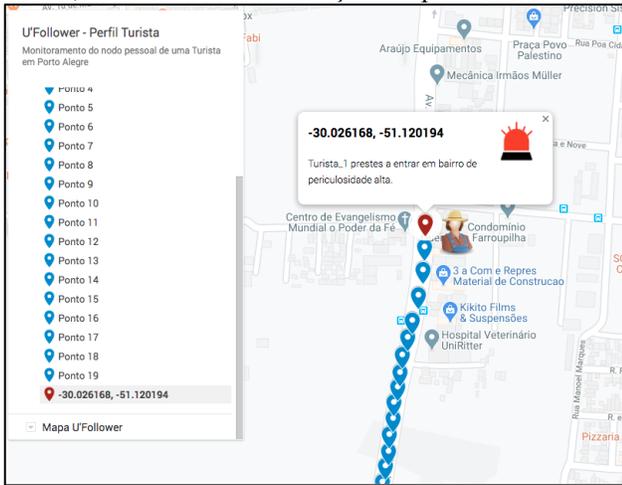


Fig. 8. Monitoramento de turista através de GPS.

Na Figura 9 é possível verificar a área limítrofe entre um bairro considerado de baixo risco pelo modelo e o bairro Rubem Berta, classificado como perigoso. No nodo pessoal da turista, a notificação aparece tanto com um texto de alerta quanto com a delimitação geográfica no formato de um polígono. Nesse contexto, toda a arquitetura do modelo é usada: o nodo pessoal da turista, a retaguarda *web* e a parte servidora.

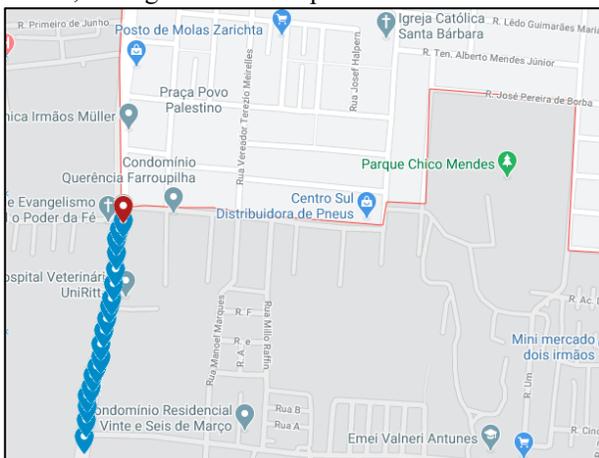


Fig. 9. Passos monitorados até a área limítrofe.

A representação da área limítrofe evidencia na parte mais clara a região onde está situado o bairro considerado de risco. Os pontos em azul representam o histórico de contexto da turista se deslocando. Por fim, o ponto em vermelho representa o local exato onde a turista estava quando recebeu a notificação, mostrando que o modelo enviou a notificação antes da turista adentrar uma região classificada como perigosa.

O modelo UFollowerServer sugeriu o uso de transporte público ou privado para sair daquela região. De forma semelhante ao aplicado no cenário 2, este cenário utilizou de forma inédita, comparando-se com os trabalhos relacionados, histórico de contextos, bem como a geolocalização, ambiente inteligente simulado e uma biblioteca geoespacial para

promover o acompanhamento ubíquo do turista.

## V. CONCLUSÃO

Este trabalho propôs o modelo UFollower, o qual contribui com a segurança nas cidades inteligentes, atuando sobre os parâmetros de coleta vindos do meio ou do próprio usuário, baseado em seu perfil. Os históricos de contextos [4] baseados em geolocalização são analisados, permitindo uma coleta otimizada e adaptável, através de abordagens de Agentes e notificações aos usuários que estejam em situações de risco. A contribuição científica do UFollower consiste em oferecer uma vigilância ubíqua assistida, otimizada pela avaliação de históricos de contextos. Este é o único modelo que utiliza concomitantemente históricos de contextos, perfis de usuários e inferência baseada em modelos probabilísticos conhecidos.

Na 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 tornazeleiras eletrônicas, bem como a avaliação das incidências de violência doméstica analisadas por Redes Bayesianas. Dessa forma, usuário e retaguarda do modelo podem tomar decisões frente ao que lhes é notificado, evitando que alguns dos índices apresentados nesse trabalho pudessem ser incrementados. A infraestrutura utilizada na simulação é conhecida e amplamente utilizada, o que não torna o projeto caro. No entanto, a bateria dos dispositivos é um fator limitante, pois o modelo não conseguirá efetuar as ações propostas no caso de falta de bateria.

Apesar da avaliação ter sido usada para situações reais de deslocamento de nodos pessoais, o UFollower pode ser usado em diferentes situações em que seja necessária a identificação de situações de risco voltadas à segurança pública.

Futuramente será necessário estudar as possibilidades de escalabilidade do modelo para que múltiplos usuários utilizem a solução, bem como avaliar a performance das soluções apresentadas (aplicativo e retaguarda *web*). A coleta de informações de *ShotSpotters* espalhados pela cidade também é uma oportunidade que ficou em aberto. Estratégias utilizando *Blockchain* podem ser exploradas para conciliar segurança nas transações de dados realizadas pelo modelo e privacidade dos usuários [21]. Por fim, abordagens de processamento de vídeo em tempo real podem ser incluídas no modelo para ampliar as possibilidades de vigilância, como, por exemplo, em acidentes de trânsito que ocorrem devido ao consumo de álcool [22].

## AGRADECIMENTOS

Os autores agradecem à Fundação de Amparo à Pesquisa do Estado do Rio Grande do Sul (FAPERGS), ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e à Universidade do Vale do Rio dos Sinos (Unisinos) pelo apoio ao desenvolvimento deste trabalho.

## REFERÊNCIAS

- [1] P. J. Navarathna and V. P. Malagi, "Artificial Intelligence in Smart City Analysis," in *2018 Int. Conf. on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2018, pp. 44–47.
- [2] G. Piro, I. Cianci, L. A. Grieco, G. Boggia, and P. Camarda, "Information centric services in Smart Cities," *J. of Systems and Software*, vol. 88, no. 10, pp. 169–188, 2014.
- [3] V. Lisboa, "Violence atlas reveals homicide rate in Brazil up 4.2%," *Agência Brasil*, Accessed on: Sep. 09, 2019, [Online] Available: <http://agenciabrasil.ebc.com.br/en/direitos-humanos/noticia/2019-06/violence-atlas-reveals-homicide-rate-brazil-42>
- [4] A. Filippetto, R. Lima, and J. Barbosa, "A Risk Prediction Model for Software Project Management based on Similarity Analysis of Context Histories," *Information and Software Technology*, vol. 106497, pp. 1–37, 2020.
- [5] Y. Pang, L. Zhang, H. Ding, Y. Fang, and S. Chen, "SPATH: Finding the Safest Walking Path in Smart Cities," *IEEE Trans. on Vehicular Technology*, vol. 68, no. 7, pp. 7071–7079, July 2019.
- [6] J. Ferreira, J. Visintin, J. Okamoto, and C. Pu, "Smart services: A case study on smarter public safety by a mobile app for University of São Paulo," in *IEEE SmartWorld Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People and Smart City Innovation*, San Francisco, CA, USA, 2017, pp. 1–5.
- [7] X. Zhao, N. Wang, R. Han, B. Xie, Y. Yu, M. Li, J. Ou, "Urban infrastructure safety system based on mobile crowdsensing," *Int. J. of Disaster Risk Reduction*, vol. 27, pp. 427–438, 2018.
- [8] O-L-M. Batista, T. Fariniuk, S. Mello, "Smart surveillance em aplicações recentes no Brasil: um estudo de caso nas cidades de Recife e Curitiba," *Revista Gestão e Secretariado - Management and Administrative Professional Review*, vol. 7, no. 2, pp. 104–137, 2016.
- [9] V. Babanne, N. Mahajan, R. Sharma, P. Gargate, "Machine learning based Smart Surveillance System," in *3rs Int. Conf. on IoT in Social, Mobile, Analytics and Cloud*, Palladam, India, 2019, pp. 84–86.
- [10] K. Muchtar, F. Rahman, M. Munggaran, A. Dwiyanoro, R. Dharmadi, I. Nugraha, "A unified smart surveillance system incorporating adaptive foreground extraction and deep learning-based classification," in *Int. Conf. on Artificial Intelligence in Information and Communication*, Okinawa, Japan, 2019, pp. 302–305.
- [11] A. Khudhair and R. Ghani, "IoT Based Smart Video Surveillance System Using Convolutional Neural Network," in *6th Int. Eng. Conf. "Sustainable Technology and Development" (IEC)*, Erbil, Iraq, 2020, pp. 163–168.
- [12] A. Dey, G. Abowd, and D. Salber, "A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications," *Human-Comput. Interact.*, vol. 16, no. 2, pp. 97–166, Dec. 2001.
- [13] A. Onisko, M. J. Druzdzel, and R. M. Austin, "Application of Bayesian network modeling to pathology informatics," *Diagnostic Cytopathology*, vol. 47, no. 1, pp. 47–47, 2019.
- [14] L. Padgham and M. Winikoff, *Developing Intelligent Agent Systems: A Practical Guide*, vol. 13, UK: John Wiley & Sons, 2005.
- [15] S. Deshmukh, D. Mane, and A. Retawade, "Building a single page application Web front-end for e-learning site," in *3rd Int. Conf. on Computing Methodologies and Communication*, Erode, India, 2019, pp. 985–987.
- [16] F. Aslam, H. Mohammed, P. Lokhande, "Efficient Way of Web Development Using Python and Flask," *Int. J. of Advanced Research in Computer Science*, vol. 6, no. 2, pp. 54–57, 2019.
- [17] H. Deitel and P. Deitel, *Java: como programar*, 8th ed., Brasil: Prentice-Hall, 2010.
- [18] M. Mannino and A. Abouzied, "Is this real? Generating synthetic data that looks real," in *32nd Annual ACM Symp. on User Interface Software and Technology*, New Orleans, USA, 2019, pp. 549–561.
- [19] M. Alawairdhi and E. Aleisa, "A Scenario-Based Approach for Requirements Elicitation for Software Systems Complying with the Utilization of Ubiquitous Computing Technologies," in *IEEE 35th Annual Comp. Software and Applications Conf. Workshops*, Munich, Germany, 2011, pp. 341–344.
- [20] E. Torres, "Cinco bairros concentram mais de um terço dos assassinatos de Porto Alegre em sete anos," *Gaúcha ZH*, Accessed on: Mar. 29, 2020, [Online] Available: <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>
- [21] O. B. Mora, R. Rivera, V. M. Larios, J. R. Beltrán-Ramírez, R. Maciel and A. Ochoa, "A Use Case in Cybersecurity based in Blockchain to deal with the security and privacy of citizens and Smart Cities Cyberinfrastructures," in *IEEE Int. Smart Cities Conf. (ISC2)*, Kansas City, USA, 2018, pp. 1–4.
- [22] A. H. Aguilar, J. C. Bonilla-Robles, J. C. Z. Díaz, and A. Ochoa, "Real-time video image processing through GPUs and CUDA and its future implementation in real problems in a Smart City," *Int. J. of Combinatorial Optimization Problems and Informatics*, vol. 10, no. 3, pp. 33–49, 2019.



**Fábio Viegas** é técnico em Sistemas de Informação pela Universidade Federal do Rio Grande do Sul (UFRGS), realizou tecnólogo em Análise de Sistemas pela Universidade do Vale do Rio dos Sinos (Unisinos). Possui especialização em Engenharia de Software na UNISC (Universidade de Santa Cruz do Sul) e mestrado em computação aplicada na Unisinos. Atualmente é analista, onde atua no mercado de trabalho desde o ano de 2010.



**Jorge Barbosa** possui graduação em Informática e Engenharia Elétrica pela Universidade Católica de Pelotas (UCPel). Ele obteve especialização em Engenharia de Software (UCPel) e concluiu mestrado e doutorado em ciência da computação na Universidade Federal do Rio Grande do Sul (UFRGS). Em 2011 realizou pós-doutorado na Sungkyunkwan University (SKKU, Suwon, Coréia do Sul). Em 2020 realizou um segundo pós-doutorado na University of California – Irvine (UCI, Irvine, EUA). Atualmente é Professor Titular II na Universidade do Vale do Rio dos Sinos (Unisinos). Ele atua como professor permanente no Programa de Pós-graduação em Computação Aplicada (PPGCA) e como professor colaborador no Programa de Pós-graduação em Engenharia Elétrica (PPGEE). Jorge coordena o Laboratório de Computação Móvel e Ubíqua (MobiLab) e atua como Bolsista de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora (bolsa DT - atualmente no Nível 1C) do CNPq. Jorge foi indicado pelo Conselho Deliberativo (CD) do CNPq para compor o Comitê de Assessoramento de Bolsas de Produtividade em Desenvolvimento Tecnológico e Extensão Inovadora (CA-DT).



**Rafael Kunst** é professor e pesquisador do Programa de Pós-Graduação em Computação Aplicada da Universidade do Vale do Rio dos Sinos (Unisinos), Brasil, onde também é membro do Laboratório de Inovação de Software - SOFTWARELAB. Ele também é

consultor ad-hoc do Ministério da Educação do Brasil. Ele possui Doutorado e Mestrado em Ciência da Computação, ambos recebidos pela Universidade Federal do Rio Grande do Sul (UFRGS). Seus interesses atuais de pesquisa envolvem comunicações móveis de última geração, como 5G e 6G, comunicações militares, Indústria 4.0, Internet das Coisas e a aplicação de Big Data Analytics e Machine Learning para otimizar as telecomunicações. Ele tem vasta experiência como consultor, coordenando e participando de projetos com universidades e empresas no Brasil e no exterior.



**Weslei Felipe Heckler** possui graduação em Ciência da Computação pela Universidade Feevale. É mestrando no Programa de Pós-Graduação em Computação Aplicada (PPGCA) da Universidade do Vale do Rio dos Sinos (Unisinos). Suas áreas de interesse são Data Science, Machine Learning e Tecnologia aplicada à Saúde.