

Online Monitoring For Public Transport Using Mobile Applications

V. C. Castro, J. S. Sá, M. A. R. Mocbel, D. Acatauassu, F. J. B. Barros and F. S. Farias

Abstract— Combined with the increasing density of mobile telephony in Brazil, the monitoring of bus routes and alert of possible dangers becomes a reality. In this context, a monitoring tool can support the safety of users and qualitatively improves the control of the location of collective vehicles. This work describes and exposes the relevance of using the system called Security System for Bus (SSBus). The SSBus consists of two pieces of software that were produced for Android and a web management system. The efficiency of this proposal was evaluated through experiments carried out in three Brazilian cities that use the third and fourth generation mobile communications networks. During the tests data were collected to generate statistics based on the precision of the implemented algorithm.

Keywords— Software development, online monitoring, mobile application, public transportation.

I. INTRODUÇÃO

A tecnologia da informação e comunicação (TIC) está ganhando um espaço cada vez maior no cotidiano das pessoas [1]. Segundo dados da Agência Nacional de Telecomunicações (ANATEL) o Brasil registrou, em dezembro de 2015, 257.79 milhões de linhas ativas na telefonia móvel e teledensidade de 125.66 acessos por 100 habitantes [2]. Na mesma direção, a cada dia surgem novos aplicativos para dispositivos móveis, dentre eles, destacam-se as soluções para a área do transporte público, tais como *softwares* para auxiliar na mobilidade urbana.

Na literatura existem diversos trabalhos que utilizam as TIC com objetivo de melhorar a gestão do serviço de transporte urbano, por exemplo, em [3] os autores buscam identificar os requisitos e atributos necessários para o desenvolvimento de um equipamento de monitoramento de ônibus que respeite as necessidades e desejos dos operadores e do órgão gestor, enquanto que, em [4] é proposto um sistema de informações de horários para usuários e administradores de transporte coletivo urbano. Já em [5] são propostas diretrizes para o desenvolvimento de um aplicativo baseado no uso do sistema global de posicionamento (*global positioning system* - GPS) que visa auxiliar ciclistas na realização da tarefa de encontrar informações que facilitem sua locomoção de forma segura e rápida.

Além de soluções relacionadas a melhoria da gestão do serviço de transporte urbano, diversos trabalhos têm proposto soluções baseadas em *hardware* e *software*. Em [6], os autores propuseram uma solução baseada em sistemas embarcados que visa gerir e monitorar a localização de ônibus a partir do GPS para melhorar o fluxo de pessoas, enquanto que os autores em [7][8][9][10][11] propuseram *softwares* para plataformas

móveis que também usam serviços de geolocalização para capturar a localização dos veículos.

Em termos de *softwares*, os autores em [7] propõem uma solução para determinação automática de ambos: rota realizada pelos ônibus e hora de chegada do veículo nos próximos pontos de ônibus. Já em [8] é proposto o *software* denominado Ônibus PoA, que visa facilitar o acesso dos usuários às informações das linhas de ônibus, como, horários e pontos de parada. Por fim, em [9],[10], e [11] os autores apresentam protótipos de aplicativos para plataforma Android com as respectivas características: o primeiro apresenta em um mapa o itinerário de uma determinada linha de ônibus selecionada, o segundo propõe uma solução de previsão de chegada de ônibus usando um sistema *online*, e o terceiro propõe um sistema de rastreamento de trânsito cooperativo baseado em *smartphones* que objetiva reduzir o tempo de espera dos usuários e a imprevisibilidade da chegada dos veículos em seus respectivos nos pontos de ônibus.

Embora as pesquisas realizadas de [3]-[11] apresentem grandes contribuições para o problema referente à pouca ou ausente informação sobre linhas de ônibus e suas respectivas rotas, seus produtos ainda não fornecem aos usuários uma solução que reúna as funcionalidades como informações sobre itinerários, acompanhamento *online* das rotas dos ônibus e principalmente a capacidade de trazer segurança para os usuários. Diante disso e no melhor de nosso conhecimento, este é o primeiro trabalho que visa ambos:

- Mostrar a viabilidade da utilização de *softwares* de monitoramento *online* para controle das rotas de linhas de ônibus; e
- Apresentar uma solução de segurança *online* baseada no estado atual (*normal*, *perigo*, *manutenção* ou *desativado*) do veículo.

Para atender os objetivos propostos, este trabalho apresenta o projeto *Security System For Bus* (SSBus). O projeto SSBus é composto pelos *softwares* SSBus Cliente e SSBus Ônibus, e por um sistema *web* de gerenciamento e armazenamento de dados. Os produtos desenvolvidos foram testados em campo com o intuito de avaliar o índice de precisão e velocidade de comunicação entre o *software* instalado no dispositivo móvel localizado no ônibus, assim como, a velocidade de aquisição das informações do *software* instalado no dispositivo do usuário. Como meio de comunicação entre os dispositivos e o servidor foi adotado a Internet móvel disponibilizada em cada cidade onde foram realizados os testes.

V. C. Castro, Bacharel em Sistemas de Informação pela Universidade Federal do Pará (UFPA), Cametá, Pará, Brasil, vicente_soad@yahoo.com.br

J. S. Sá, Bacharelando em Sistemas de Informação pela UFPA, Cametá, Pará, Brasil, joinersa@hotmail.com

M. A. R. Mocbel, Bacharel em Sistemas de Informação pela UFPA, Cametá, Pará, Brasil, angelomocbel@gmail.com

D. Acatauassu, Professor Adjunto na Faculdade de Sistemas de Informação

- UFPA, Cametá, Pará, Brasil, diogoaca@ufpa.br

F. J. B. Barros, Professor Adjunto na Faculdade de Engenharia Elétrica - UFPA, Tucuruí, Pará, Brasil, fbarros@ufpa.br

F. S. Farias, Professor Adjunto na Faculdade de Sistemas de Informação - UFPA, Cametá, Pará, Brasil, fabriciosf@ufpa.br

Corresponding author: Vicente Caldas de Castro.

II. METODOLOGIA DO PROJETO

A realização deste projeto contou com uma metodologia baseada em quatro etapas, sendo estas: (i) Pesquisa de *softwares* para plataformas móveis que utilizam mapas para auxílio de motoristas e pedestres, (ii) Levantamento de requisitos e criação do projeto, (iii) Implementação e (iv) Testes e análise dos resultados.

Na primeira etapa, foram pesquisados aplicativos, na plataforma virtual da Google Play, que utilizam mapas para auxiliar seus usuários, tanto motoristas quanto pedestres, em relação ao trânsito. Os aplicativos encontrados foram usados como base para a construção e implementação dos *softwares* SSBus Cliente e SSBus Ônibus. Os principais aplicativos encontrados estão disponíveis em [12] e são listados abaixo:

- Maps GPS e transporte público: Serviço de mapas da empresa Google, disponibiliza ao usuário a busca de rotas para determinados destinos, agrega informações de locais e pontos, como também disponibiliza de forma *online* informações de trânsito e transporte público;
- Uber: Aplicativo que permite ao usuário contratar viagens rápidas, acompanhando *online* os carros disponíveis para serviço;
- Waze: Aplicativo de navegação colaborativo em que os próprios usuários compartilham *online* informações sobre trânsito e rotas;
- HERE WeGo: Aplicativo que fornece um mapa com funcionalidades de navegação por voz para vários tipos de transporte, gerando a melhor rota;
- Strava GPS Correr Ciclismo: Oferece serviços de geolocalização para pedestres e ciclistas informando quais os melhores pontos para fazer passeios ou exercícios;
- Moovit: Disponibiliza aos usuários, itinerários de ônibus, trens e metrô cadastrados no serviço, reunindo todas as informações para ajudar o usuário a utilizar o transporte público;
- Cittamobi: Permite aos usuários acompanhar o transporte público de sua cidade, exibindo *online* os ônibus no mapa, no qual é possível buscar pontos e linhas de ônibus e consultar o tempo que o ônibus leva para chegar em um ponto.

Na segunda etapa, foram levantados os requisitos para o sistema, usando como referência os aplicativos pesquisados na primeira etapa, buscando maneiras de melhorar os *softwares* já existentes com o intuito de garantir melhor qualidade de experiência aos usuários. Nesta etapa, também foi realizada a criação do projeto e modelagem do *software*, aplicando o padrão Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML) [13]. Foram implementados os seguintes diagramas: Caso de Uso, de Classe e de Estado.

Na terceira etapa, após a modelagem do projeto, iniciou-se a fase de implementação dos aplicativos e do sistema de gerenciamento. Para a criação dos aplicativos, foram utilizadas as linguagens de programação Java e *Extensible Markup Language* (XML), em conjunto com o ambiente de programação Android Studio. Para a criação do site responsável em realizar a comunicação com o servidor de banco de dados foram utilizadas as linguagens *Structured Query Language*

(SQL) e PHP: *Hypertext Preprocessor* (PHP), ambas usando o *software* NetBeans. Já para a criação do componente de gerência, foram utilizadas as linguagens *HyperText Markup Language* (HTML), para criação dos layouts das páginas *web* responsáveis em cadastrar, pesquisar e excluir os ônibus, e PHP para a construção da página e comunicação com o servidor de banco de dados.

Após a criação dos dois *softwares* e da página *web* de gerenciamento, foi realizada a etapa de experimentos. Os experimentos foram realizados em 3 (três) cidades brasileiras, sendo elas: Belém, Cametá e Tucuruí.

Para avaliação dos testes, foram criadas métricas para medir o desempenho dos *softwares*, uma delas é a métrica que usa o tempo de resposta da comunicação entre os aplicativos e o servidor de dados localizado na nuvem. Durante os testes foram utilizados 2 *smartphones*. O *software* SSBus Ônibus foi instalado no *smartphone* que ficou dentro do veículo, enquanto que o aplicativo SSBus Cliente foi instalado e monitorado por um usuário fora do veículo.

III. PROJETO SECURITY SYSTEM FOR BUS

O projeto *Security System For Bus* (SSBus) propõe o desenvolvimento e utilização de dois *softwares mobile* que visam realizar o monitoramento *online* da geolocalização e o estado atual, que podem ser, *normal*, *perigo*, *manutenção* ou *desativado*, dos ônibus cadastrados no sistema. A Fig. 1 apresenta a definição da arquitetura do projeto, a qual é composta por um serviço baseado no envio e consulta de informações em um determinado servidor de banco de dados.

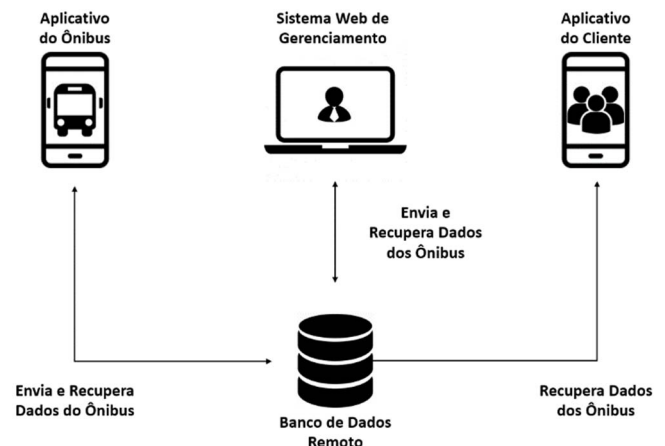


Figura 1. Definição da arquitetura do projeto.

O aplicativo SSBus Ônibus, também denominado *aplicativo do ônibus*, é responsável em rastrear, via GPS, e enviar as coordenadas de latitude e longitude do veículo para o servidor de banco de dados localizado na nuvem. O aplicativo também é responsável em manter o estado atual do ônibus atualizado no servidor. Ambos os serviços são transmitidos e recebidos no servidor de banco de dados usando o serviço de Internet de 3ª ou 4ª geração (3G ou 4G) disponibilizados por operadoras de telefonia móvel. A Fig. 2 ilustra a tela principal do *aplicativo do ônibus*.



Figura 2. Telas principais do aplicativo do ônibus.

As informações armazenadas no servidor localizado na nuvem são recuperadas pelo aplicativo SSBUS Cliente, também denominado *aplicativo do cliente*. Este aplicativo é responsável em acessar o servidor de banco de dados para realizar a coleta das seguintes informações: coordenadas geográficas e estado atual de determinada linha de ônibus. Assim como o *aplicativo do ônibus*, este aplicativo necessita de conexão com a Internet para garantia da comunicação com o servidor. Na sequência do texto detalharemos os elementos de *software* que compõem o projeto. A Fig. 3 ilustra a tela principal do *aplicativo do cliente*.



Figura 3. Tela principal do aplicativo do cliente.

Aplicativo do ônibus

Conforme informado, o *aplicativo do ônibus* é responsável por coletar coordenadas de geolocalização e o estado do ônibus.

As informações de localização são adquiridas a partir da utilização do GPS nativo do celular, enquanto que o estado do ônibus é definido como *normal* por padrão, mas pode ser editado pelo usuário para os estados de *desativado*, *manutenção* ou *perigo*.

Conforme a Fig. 2, a tela principal do *aplicativo do ônibus* é organizado da seguinte forma: três botões de interação, um menu localizado no lado superior direito da tela, e uma tela informativa que indica o nome do ônibus e o seu código de placa, por exemplo, Linha Amarela (JFR – 0245).

Os três botões de interação são, respectivamente, responsáveis por modificar o estado para *manutenção* ou *perigo*, e para selecionar a linha de ônibus que irá sair para fazer a sua rota. Quando um dos botões de estado é pressionado, uma atualização do estado atual do ônibus é enviada para o servidor. Esta atualização é imediatamente disponibilizada a todos os usuários que, *online*, verificam a localização dos ônibus em seu aplicativo. Já o botão selecionar ônibus permite ao motorista escolher de forma genérica um dos ônibus cadastrados no banco de dados, ou seja, o aplicativo permite reutilização dos celulares, pois o mesmo pode ser utilizado, conforme a necessidade, em diferentes ônibus. Após a escolha, o veículo aparece ativo e disponível no mapa dos usuários.

Por último, ao pressionar o menu localizado no lado superior direito da tela (vide Fig. 2) são apresentadas as seguintes funcionalidades: desativar emergência, desconectar ônibus e o sobre (apresenta as informações dos envolvidos no projeto). Na funcionalidade desativar emergência, o usuário pode reconfigurar o estado atual para normal, por exemplo, caso o ônibus esteja recebendo manutenção e esta seja concluída, o motorista poderá informar aos usuários alterando o estado atual. Enquanto que, a funcionalidade desconectar ônibus permite ao usuário transferir e associar o celular com outro ônibus.

Aplicativo do cliente

O *aplicativo do cliente* é responsável em recuperar e apresentar as informações armazenadas no servidor de banco de dados. As informações são filtradas a partir do interesse dos usuários do serviço, isto é, o usuário pesquisa a partir do nome da linha de ônibus desejada.

A Fig. 3 apresenta a tela principal do *aplicativo do cliente*. O aplicativo é organizado da seguinte forma: botão de interação na tela inicial e um menu superior que contém as configurações gerais do aplicativo.

O botão de interação *ir para o mapa* busca no servidor as coordenadas de latitude e longitude, e o estado atual dos ônibus. Em seguida, as informações são apresentadas ao usuário em um mapa. Os ônibus são apresentados no mapa usando os marcadores padrões do Google Maps, o qual é uma API nativa da plataforma Android. Os estados atuais são simbolizados por cores, onde:

- Verde: significa o estado *normal*;
- Amarelo: simboliza o estado de recebimento ou espera de *manutenção*;
- Vermelho: significa o estado de *perigo*;
- Azul claro: simboliza o estado *desativado*;
- Azul escuro: representa a localização do usuário.

No menu são apresentados aos usuários as seguintes opções: *selecionar ônibus*, *excluir ônibus*, *receber coordenadas do*

ônibus e *sobre* (apresenta as informações dos envolvidos no projeto). Na opção de *selecionar ônibus* é apresentada uma lista com todas as opções de ônibus cadastradas no servidor. O usuário pode selecionar as linhas de interesse, assim definindo suas linhas de ônibus favoritas e quais devem ser apresentadas no mapa. A funcionalidade de *excluir ônibus* possibilita ao usuário eliminar determinada opção que deixe de ser favorita. Por último, a funcionalidade *receber coordenadas do ônibus* permite a configuração da taxa de recebimento de dados a partir do servidor, por exemplo, de cinco em cinco segundos. Esta funcionalidade permite ao usuário economizar a sua franquia de Internet.

IV. MÉTRICAS DE AVALIAÇÃO

Para validar a viabilidade do projeto SSBUS como uma ferramenta *online* de monitoramento de rotas e segurança, foram criadas as métricas de avaliação da porcentagem de erro por rota ($P_{e/rota}$), tempo médio entre os saltos (t_{salto}) e percentual de tempo de erro (P_{erro}). As métricas de porcentagem avaliam a eficiência na tarefa de monitoramento, enquanto que, a métrica de tempo avalia o desempenho do sistema como uma ferramenta de segurança.

A primeira métrica, (equação (1)), foi criada para medir o percentual de erro na rota feita pelos ônibus, ou seja, quantas vezes o marcador se apresenta fora da via (rua, avenida, rodovia, etc.) no mapa.

$$P_{e/rota} = \frac{N_e}{(N_a + N_e)} \times 100 \quad (1)$$

onde, N_e é o números de saltos para fora da via e N_a são os saltos corretos.

A segunda métrica é responsável em calcular o tempo médio, em segundos, entre os saltos durante a execução da rota, conforme mostra equação (2).

$$t_{salto} = \frac{T_{tot}}{(N_a + N_e)} \quad (2)$$

onde, T_{tot} é o tempo total, em segundos, que o ônibus levou para completar a rota.

Por fim, a métrica 3 é responsável em medir o percentual de tempo que o marcador do ônibus foi apresentado no mapa em uma localização fora da via, vide a equação (3).

$$P_{erro} = \frac{\sum_i^{N_e} T_e(i)}{T_{tot}} \times 100 \quad (3)$$

onde, $T_e(i)$ é o tempo, em segundos, em que o ônibus esteve fora da via no salto i .

V. TESTES E RESULTADOS

Durante o projeto foram realizados cinco testes de campo. O objetivo dos testes foi validar a eficiência dos *softwares* na realização de sua tarefa de gerenciamento dos ônibus.

Para a realização dos testes foram utilizados dois celulares que representam o sistema do cliente e o sistema do ônibus. Para transporte das informações (*uplink* e *downlink*) foi utilizada a conexão de Internet 3G e 4G das operadoras de telefonia móvel disponíveis nos municípios. Os experimentos

foram realizados nas cidades de Cameté, Tucuruí e Belém – todas localizadas no Estado do Pará. Para a realização dos testes foram utilizados veículos particulares, os quais representam os ônibus nesse manuscrito.

Para melhor análise dos locais de teste, foram identificados os locais das torres de transmissão das redes móveis a partir do aplicativo móvel da ANATEL [14].

Primeiro teste – cidade de Cameté

O primeiro experimento foi realizado nos bairros centrais da Marambaia, Centro e Matinha. Estes bairros concentram grande parte da população da cidade. A distância percorrida entre a posição inicial e salto final foi de 2.44 quilômetros, conforme descrito na Fig. 4. Durante a execução do experimento foi obtido a velocidade média de 31 km/h. Para comunicação entre os *softwares* e servidor foi utilizado a tecnologia de comunicação de dados de terceira geração.



Figura 4. Rota feita pelo ônibus na cidade de Cameté.

Segundo teste – cidade de Tucuruí

O segundo experimento foi realizado em regiões da cidade de Tucuruí que compreendem os bairros Jardim Marilucy, Vila Permanente e Colinas. Desta vez, a distância total percorrida foi de 3.63 quilômetros e a velocidade média alcançada foi de 25 km/h. A Fig. 5 ilustra a rota percorrida. Para comunicação entre os *softwares* e servidor foi utilizado a tecnologia de comunicação de dados de terceira geração.



Figura 5. Rota feita pelo ônibus na cidade de Tucuruí.

Terceiro teste – Vila Permanente de Tucuruí

O terceiro também foi realizado na Cidade de Tucuruí, mas desta vez, foi realizado exclusivamente no bairro vila permanente. Durante o experimento foi percorrido a distância de 5,8 quilômetros com velocidade média de aproximadamente 33 km/h. A Fig. 6 ilustra a rota percorrida, a qual fez uso da tecnologia de comunicação de dados de terceira geração.

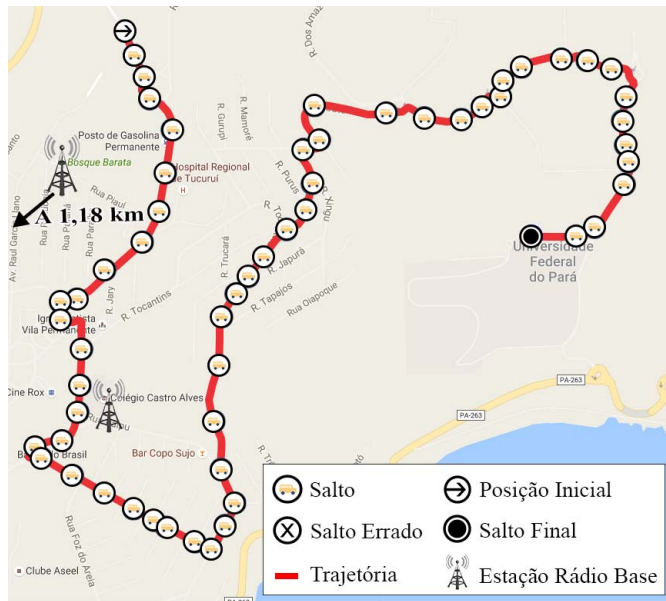


Figura 6. Rota realizada no segundo teste na Vila Permanente (Tucuruí).

Quarto teste – Belém do Pará

O quarto experimento foi realizado nas proximidades do Campus Universitário da UFPA, Cidade de Belém. Neste teste foram percorridos os bairros do Curió Utinga e Guamá. Desta vez, a velocidade média foi de 31,2 km/h e a distância percorrida foi de 8,1 quilômetros, a qual foi a maior entre todos os experimentos. A Fig. 7 apresenta a rota percorrida. Além de representar a maior distância percorrida, este experimento fez uso da tecnologia de comunicação 4G, disponível na Capital do Estado.

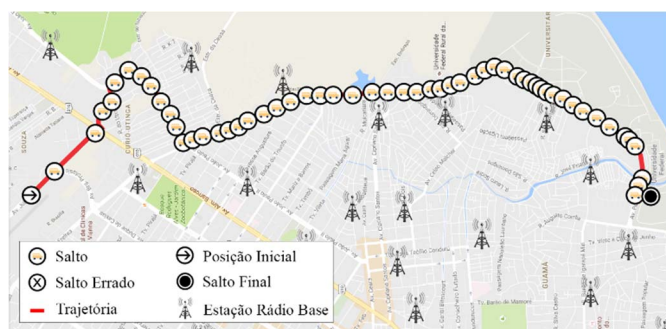


Figura 7. Teste realizado na cidade de Belém.

Quinto teste – cidade universitária José Silveira Neto

O último experimento foi realizado na cidade Universitária José Silveira Neto, localizado em Belém. Neste teste, foi percorrida a rota do ônibus que transporta estudantes dentro da Cidade Universitária. Desta vez, foram percorridos 7,16

quilômetros com uma velocidade média de 29 km/h, ilustrado na Fig. 8. O experimento também fez uso da tecnologia de comunicação de dados de quarta geração, disponível na Capital do Estado.



Figura 8. Teste realizado na cidade Universitária José Silveira Neto.

Resultados

Após a realização dos testes, os dados foram coletados a partir do *aplicativo do cliente* e utilizados nas métricas de avaliação da precisão do sistema em relação à localização e rota. A Tabela I apresenta os resultados de desempenho alcançados durante os testes.

TABELA I
RESULTADOS OBTIDOS PARA OS EXPERIMENTOS A – CAMETÁ, B – CIDADE DE TUCURUÍ, C - VILA PERMANENTE, D – BELÉM, E E – CIDADE UNIVERSITÁRIA JOSÉ SILVEIRA NETO/BELÉM.

P/R ^a	A	B	C	D	E
N _a	25	23	57	56	102
N _e	1	0	0	0	16
T _{tot}	505 s	543 s	636 s	933 s	890 s
T _{e(i)}	17 s	0 s	0 s	0 s	84 s
P _{e/rota}	4%	0%	0%	0%	13,5%
t _{salto}	20 s	23 s	11 s	17 s	7 s
P _{erro}	3,36%	0%	0%	0%	9,43%

a. P = Parâmetro/R = Rota

Nota-se que em todos os testes o N_e se manteve muito baixo, ou seja, em 280 saltos registrados foram identificados somente 17 saltos fora da rota prevista, assim, levando a uma média global de acertos de 93,92%.

A segunda métrica analisada pode influenciar em diversas decisões do usuário como, por exemplo, quando se deslocar para o ponto de ônibus ou sobre a tomada de decisão referente a determinada modificação do estado do ônibus para *perigo* ou *manutenção*. Considerando os cinco experimentos, o tempo médio global entre os saltos foi de 12,5 segundos, além disso, o tempo máximo de espera entre o salto e apresentação da informação ao usuário do *software* SSBUS Cliente foi de 23 segundos.

Vale salientar que as métricas utilizadas neste trabalho são válidas para confirmar a funcionalidade do *software* em termos de geolocalização correta e tempo de resposta. Estes termos estão relacionados com a latência de resposta do tipo de rede móvel considerada (3G ou 4G), e o tipo de relevo ao redor das rotas. No entanto, mesmo com as limitações de rede e diferentes tipos rotas consideradas, os *softwares* conseguem apresentar suas funcionalidades em intervalos de segundos, por exemplo, o tempo máximo de salto demonstra que em caso de uma

situação de modificação de estado para *perigo*, as autoridades competentes podem receber as informações em um tempo menor ou igual a 20 segundos.

Outro resultado relevante está associado ao consumo de dados da franquia móvel para operação dos *softwares*. A cada uma hora de uso, o *aplicativo do ônibus* consome em média 5.3 MB, enquanto que, o *aplicativo do cliente* consome 2.6 MB. A diferença de consumo de dados entre os *softwares* é justificada pela maior taxa de atualização do *aplicativo do ônibus*, pois este é responsável em atualizar as coordenadas no servidor dados sempre que o veículo se locomove. Por fim, este resultado demonstra o baixo impacto no consumo de Internet.

VI. CONCLUSÃO

Este trabalho apresentou os resultados alcançados durante a execução do projeto *Security System For Bus*, o qual foi desenvolvido com objetivo de auxiliar os usuários do transporte público, assim como na tentativa de inibir ações criminosas, como assaltos, em ônibus. Durante o projeto foram desenvolvidos e testados dois *softwares* disponibilizados para o sistema operacional Android. Os aplicativos foram testados em três cidades do norte do Brasil: Cametá, Tucuruí e Belém. Os resultados comprovam a eficiência da aplicação na realização da tarefa de identificação *online* da localização dos ônibus. Além disso, a solução proposta demonstrou precisão média de acertos superior a 93.9%.

Os resultados mostram que a ferramenta proposta tem potencial para auxiliar o monitoramento *online* da segurança pública usando o compartilhamento do estado atual (*normal*, *manutenção* e *perigo*) do ônibus como alternativa de informação. Esta informação pode auxiliar na melhoria da segurança, via estímulo de denúncias por parte dos usuários do aplicativo, por exemplo, em caso de modificação do estado para *perigo*, ou até mesmo via adesão do poder público à utilização dos aplicativos como suporte às centrais de inteligência da polícia civil e militar. Em todos os experimentos o tempo de alteração de um estado para o outro se manteve igual ou inferior a 20 segundos.

A partir deste trabalho, conclui-se a relevância da utilização dos *softwares* como ferramentas de atendimento das necessidades de usuários de transportes coletivos. Por fim, as métricas apresentadas neste trabalho podem auxiliar na futura comparação entre diferentes tipos de arquiteturas de sistemas *web* desenvolvidos.

AGRADECIMENTOS

Os autores agradecem a Pró-Reitoria de Extensão - UFPA pelo apoio e suporte financeiro durante o período de 2015 a 2017 (Vigência de projetos de extensão financiados).

REFERÊNCIAS

- [1] D. A. Nóbrega, G. Fenner and A. S. Lima, "A Risk Management Methodology Proposal for Information Technology Projects", in *IEEE Latin America Transactions*, vol. 12, no. 4, pp. 643-656. 2014.
- [2] IDGNOW!, "Telefonia móvel perde 22,94 milhões de linhas ativas em um ano". Available in: <<http://idgnow.com.br/mobilidade/2016/02/01/telefoniamovel-perde-22-94-milhoes-de-linhas-ativas-em-um-ano/>>. 2016.

- [3] D. M. Silva, "Sistemas Inteligentes no Transporte Público Coletivo por Ônibus", Dissertação de Mestrado, Escola de Engenharia, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil, 2000.
- [4] L. Weigang, Y. Yamashita, O. Q. Silva, D. XiJun, M. A. T. Prazeres, D. C. S. Oliveira, "Implementação do Sistema de Mapeamento de uma Linha de Ônibus para um Sistema de Transporte Inteligente", Anais do XXI Congresso da Sociedade Brasileira de Computação, Seminário Integrado de Software e Hardware (SEMISH). 2001.
- [5] E. Cesani, L. Dranka, "Diretrizes para desenvolvimento de Aplicativo GPS em dispositivos móveis para ciclistas de Curitiba", *InfoDesign-Revista Brasileira de Design da Informação*, v. 9, n. 2, pp. 123-138, 2013.
- [6] E. S. Souza, J. d. C. Barbosa, F. M. Millian, M. Torres and P. E. Ambrosio, "Tracking System for Urban Buses with People Flow Management," in *IEEE Latin America Transactions*, vol. 9, no. 6, pp. 944-949. 2011.
- [7] J. Biagioni, T. Gerlich, T. Merrifield, and J. Eriksson, "EasyTracker: automatic transit tracking, mapping, and arrival time prediction using smartphones", in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys '11)*, ACM, New York, NY, USA, pp. 68-81. 2011.
- [8] R. V. Gargione, "Plano de marketing para lançamento do APP Onibus POA", Trabalho de Conclusão de Graduação, Escola de Administração, Universidade Federal do Rio Grande do Sul, Porto Alegre, RS, Brasil, 2011.
- [9] C. C. Oliveira, "Estudo sobre a utilização de mapas em Android para itinerários de ônibus", Monografia de Especialização, Departamento Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Curitiba, PR, Brasil, 2012.
- [10] P. Zhou, Y. Zheng and M. Li, "How long to wait?: predicting bus arrival time with mobile phone based participatory sensing", in *Proceedings of the 10th international conference on Mobile systems, applications, and services (MobiSys '12)*, ACM, New York, NY, USA, pp. 379-392. 2012.
- [11] A. Thiagarajan, J. Biagioni, T. Gerlich, and J. Eriksson, "Cooperative transit tracking using smart-phones", in *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems (SenSys '10)*, ACM, New York, NY, USA, pp. 85-98. 2010.
- [12] GOOGLE. Google Play. Available in: <https://play.google.com/store/apps?hl=pt_BR>.
- [13] R. Pressman, "Engenharia de Software", 6a ed., São Paulo: Makron Books. 1056 p. 1995.
- [14] ANATEL. Anatel Serviço Móvel. Google Play: Google, 2017. Available in: <<https://play.google.com/store/apps/details?id=br.com.mais2x.anatelsm&hl=pt-BR>>.



Vicente Caldas de Castro, Bacharel em Sistemas de Informação pela UFPA – Campus Universitário do Tocantins/Cametá. Membro colaborador do Laboratório de Programação Extrema desde 2015.



Joiner dos Santos Sá, discente do curso de Sistemas de Informação da UFPA – Campus Universitário do Tocantins/Cametá. Membro colaborador do Laboratório de Programação Extrema desde 2015.



Miguel Ângelo Rodrigues Mocbel, Bacharel em Sistemas de Informação pela UFPA – Campus Universitário do Tocantins/Cametá. Membro colaborador do Laboratório de Programação Extrema desde 2015.



Diogo Lobato Acatauassú Nunes, possui graduação em Engenharia de Computação (2006) e Mestrado e Doutorado em Engenharia Elétrica (2009 e 2014), todos pela UFPA. Atualmente é Professor Adjunto I da UFPA – CUNTINS. Ministrou disciplinas na Faculdade de Engenharia de Computação da UFPA, em Belém, entre os anos de 2008 e 2011 e trabalhou como pesquisador visitante no Ericsson Research Center em Estocolmo, Suécia, em 2011.



Fabricio Jose Brito Barro, possui graduação em Engenharia Elétrica pela Universidade Federal do Pará (2003), mestrado e doutorado em Engenharia Elétrica pela Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). Atualmente é Professor Adjunto da Universidade Federal do Pará Campus de Tucuruí.



Fabricio de Souza Farias é professor adjunto vinculado à Faculdade de Sistemas de Informação da Universidade Federal do Pará (UFPA) – Campus Cametá. É engenheiro da Computação e tem mestrado e doutorado em engenharia elétrica pela UFPA. Suas áreas de pesquisa são computação aplicada a telecomunicações, inteligência artificial, mineração de dados, fundamentos de computação e engenharia de software.