

Um Sistema de Baixo Custo para Redes Inteligentes de Transporte Público

Marlos Marques

Dept. de Ciências Exatas e Tecnológica
UESB

Vitória da Conquista, Bahia, Brasil
marlos@uesb.edu.br
ORCID 0000-0003-3782-5543

Cássio Silva

Dept. de Ciências Exatas e Tecnológica
UESB

Vitória da Conquista, Bahia, Brasil
cassiomeirasilva12@gmail.com
mORCID 0000-0002-2369-5891

Jorge Correia

Dept. de Ciências Exatas e Tecnológica
UESB

Vitória da Conquista, Bahia, Brasil
correia.jorge@protonmail.com
ORCID 0000-0002-6035-7314

Resumo - O Sistema de Transporte Público (STPU) é utilizado por uma parcela razoável da população, além de ser considerado como forma de deslocamento urbano sustentável e escalável. Apesar disso, aplicativos que atendam satisfatoriamente os usuários do STPU ainda são escassos. Em parte, esta dificuldade esta relacionada aos equipamentos necessários para prover a infraestrutura de suporte necessária a tais sistemas, principalmente as informações relacionados a geolocalização. O objetivo deste trabalho foi desenvolver um sistema completo de baixo custo que dê suporte aos usuários do transporte público, possibilitando-os obter informações sobre os itinerários, pontos de embarque e desembarque e a localização dos veículos em tempo real.

Palavras-chave: SPTU, aplicações móveis, lixo eletrônico.

I. INTRODUÇÃO

Nos últimos anos a utilização de dispositivos móveis vem crescendo cada vez mais e hoje tornou-se uma ferramenta indispensável na vida das pessoas. Aplicativos diversos facilitam o gerenciamento de tarefas cotidianas em áreas como comunicação, serviços, lazer, cidadania, entre outras. Na área de transporte privado o uso de tais aplicativos tornou-se destaque [1]. Entretanto, o Sistema de Transporte Público (STPU), que é utilizado por uma grande parcela da população, ainda não possui a mesma gama de serviços, mesmo sendo considerado uma das formas mais sustentáveis de deslocamento no ambiente urbano, por reduzir o índice de tráfego e pelo seu baixo impacto ambiental e financeiro [2].

II. TRABALHOS RELACIONADOS

Diversos problemas são enfrentados pelos usuários destes transportes, a exemplo de atrasos excessivos, ônibus superlotados e falta de segurança [3]. A solução de alguns desses problemas pode ser mitigada através do acesso rápido e descomplicado a informações sobre o próprio STPU. Neste contexto de Cidades Inteligentes, soluções deste tipo estão sendo propostas, entretanto dois aspectos merecem observação: ou ficam restritas a grandes centros urbanos, como por exemplo São Paulo, com o Citymapper [4], Urbanoide [5], Cadê o Ônibus [6], entre outros, e Rio de Janeiro, com o Vá de Ônibus [7], entre outros. Ou quando estão disponíveis em cidades de menor porte não fornecem informações verdadeiramente em tempo real, utilizando estratégias que se valem de notificações fornecidas pelos próprios usuários do sistema, como por exemplo o Moovit [8] e o CittaMobi [9].

Nesse sentido, o RITP fornece ao usuário do STPU acesso fácil e rápido a informações simples, porém imprescindíveis para um melhor planejamento do deslocamento, tais como linhas de ônibus, itinerários, pontos de embarque e desembarque e localização dos veículos em tempo real, a um baixo custo de implementação.

III. METODOLOGIA

O sistema RITP possui quatro componentes chaves: um aplicativo móvel utilizado pelo usuário (MoviBus), um módulo rastreador que fornece os dados do veículo (Tracker), um banco de dados para armazenamento de informações (RITPBase) e a geolocalização por satélites GPS. Uma visão do geral pode ser vista na Figura 1 e cada componente será detalhado a seguir.



Fig. 1. Visão geral do sistema

Para o desenvolvimento do projeto foi utilizado o padrão MVC (Model View Controller), o qual divide uma aplicação em três partes interconectadas (modelo, visão e controlador) [10] permitindo facilidades na reutilização e manutenção do código. No processo de modelagem foi utilizada a linguagem UML [11] em conjunto com a ferramenta Astah Community [12] para a elaboração e construção dos diagramas de caso de uso, de classes, de objetos, de estados, de componentes, de atividades e de sequência. A prototipagem dos aplicativos utilizou o Balsamiq Mockups [13], um software gratuito que possui um processo simples e rápido de desenvolvimento de interfaces. Por fim, o sistema foi implementado inicialmente para dispositivos Android, por se tratar da plataforma móvel com maior penetração no mercado [14] [15] utilizando o SDK de desenvolvimento Android Studio [16] [17].

A. Tracker App

Para prover informações de rastreamento em tempo real é necessário obter os dados de localização geográfica do veículo e disponibilizá-los ao usuário. O Tracker é um módulo de localização de baixo custo que disponibiliza esta informação ao RITP. Para esta função, ele utiliza aparelhos celulares antigos, o que difere de outras implementações que utilizam dispositivos dedicados ou que não fornecem dados em tempo real. Esta estratégia pode baratear o custo do RITP em até 50% [18], comparando aos sistemas convencionais que utilizam aparelhos de GPS dedicados, necessitando apenas definir qual o software/hardware mínimo aceitável para os celulares reaproveitados.

Outro fator importante está relacionado a nova destinação dada a estes aparelhos. Aspectos comuns ao descarte de celulares antigos, como por exemplo telas quebradas ou trincadas, além do esgotamento da bateria, não são relevantes ao escopo do projeto [19] [20]. A única exigência é possuam o GPS e o GPRS/2G/3G/4G funcionando. Esta característica contribui para diminuição do lixo tecnológico, uma vez que até 80% deste material no mundo é descartado sem obedecer a nenhum critério ou respeito pelo homem ou pela natureza, segundo a United Nations Environment Programme (UNEP), órgão das Nações Unidas voltado para o meio ambiente [21].

O módulo é composto pelo aplicativo em si e por um circuito conversor de tensão de 12V para 5V (Figuras 2a e 2b). O dispositivo é então instalado no veículo e interage com os outros componentes do RITP coletando informações de geolocalização via GPS e fornecendo estes dados ao sistema via rede móvel de dados (GPRS, 2G, 3G ou 4G). Além das coordenadas (latitude e longitude), estes dados contém informações sobre o veículo e a linha ao qual o Tracker transmissor pertence. Todo conjunto de informações coletadas são indexadas em um arquivo JSON (Figura 2c).

sistema de transporte, como por exemplo ônibus, linhas, itinerários, pontos de parada, além das informações dos usuários cadastrados, como por exemplo login, senha e favoritos.

O gerenciador de banco de dados escolhido foi o Firebase do Google por sua extrema versatilidade e extensa gama de recursos [22]. Dentre eles, vale ressaltar o Firebase Firestore Database e o Firebase Realtime. O Realtime Database é um banco de dados NoSQL (Not Only SQL) hospedado na nuvem que possibilita sincronização de dados em tempo real. Sua estrutura é baseada em uma árvore JSON, no qual os dados são armazenados nos nodos, facilitando a modelagem dos dados. Seu recurso de sincronização em tempo real é utilizado pelas informações geradas pelo módulo Tracker. Já o Firestore Database é orientado a documentos e possui suporte off-line, ou seja, os dados que foram carregados ficam armazenados em cache, permitindo acesso quando o dispositivo não possuir conexão com a Internet, o qual se mostrou adequado ao gerenciamento dos dados estáticos.

No diagrama de classes da Figura 3 é possível visualizar a estrutura de dados implementada no RITP, onde as principais classes são: Usuário, Ônibus, Linha, Rota, PontoOnibus, PontoFavorito e LinhaFavorita.

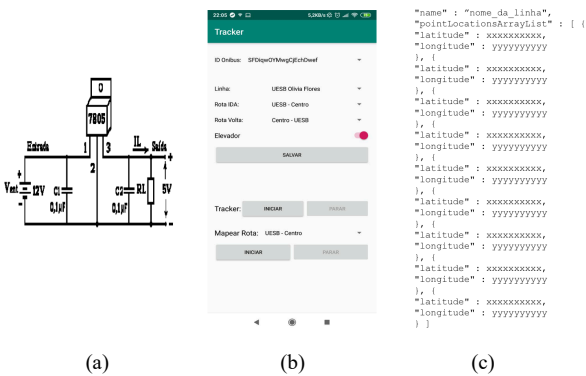


Fig. 2. Módulo Tracker

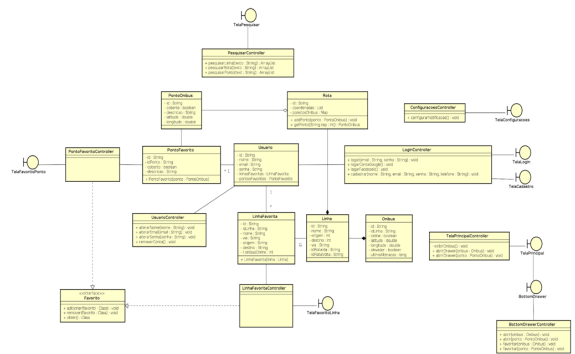


Fig. 3. Diagrama de classes

C. MoviBus App

É o aplicativo que permite ao usuário do RITP interagir com o STPU e já se encontra disponível para download na Play Store [23]. A funcionalidades previstas pelo aplicativo incluem efetuar login, visualizar próximos ônibus de um ponto, criar ponto de ônibus favorito, criar alerta de chegada de ônibus, criar alerta de descida, monitorar ônibus, visualizar rota de ônibus, criar ônibus favorito, pesquisar ônibus, pesquisar ponto de ônibus e gerenciar cadastro. Na Figura 4 é possível ter uma visão completa das possibilidades de interação com o usuário.

B. RITPBase

Os dados gerados pelo sistema são armazenados no módulo chamado RITPBase. Estes dados são classificados em duas categorias: dinâmicos e estáticos. Os chamados dinâmicos ou de tempo real, notadamente as informações de geolocalização do veículo, possuem uma grande volatilidade e necessitam de um gerenciamento diferenciado para não estressarem o sistema. Os dados estáticos ou convencionais estão relacionados as características do

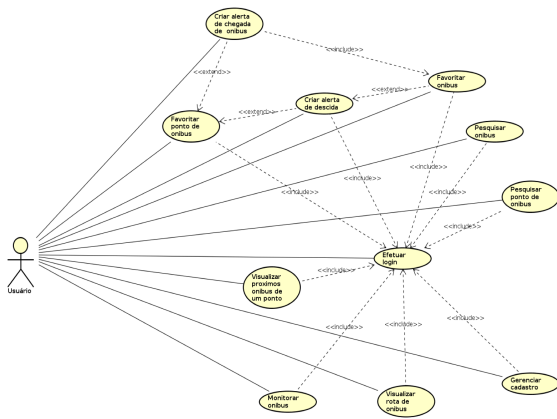


Fig. 4. Diagrama de caso de uso

Quando o usuário realiza uma ação esta se traduz em um evento no sistema. O diagrama de sequência é utilizado para representar a interação entre os objetos que colaboram entre si para realizar este evento. Na Figura 5 tem-se o diagrama de sequência para o caso de uso no qual o usuário define uma linha de ônibus como favorita. O evento acontece quando o ator usuário envia uma mensagem “abrirDrawer” para o objeto TelaPrincipalController, passando o ônibus selecionado como parâmetro. Logo, o objeto TelaPrincipalController fica responsável em chamar o objeto BottomDrawerController através da mensagem “abrir”, passando o mesmo ônibus como parâmetro. Após o retorno, com a mensagem de “tela aberta” o ator do sistema pode acessar diretamente o objeto BottomDrawerController enviando a mensagem “favoritar”. No mesmo objeto controller é criada uma nova instância para o objeto LinhaFavorita, no qual existe as informações da linha do ônibus que o usuário havia selecionado inicialmente. Com a instância do objeto LinhaFavorita criada, o mesmo é enviado para o objeto FireLinha que se comunica com o ator RITPBase, que contém o sistema de banco de dados para armazenar o valor da nova linha preferida pelo usuário.

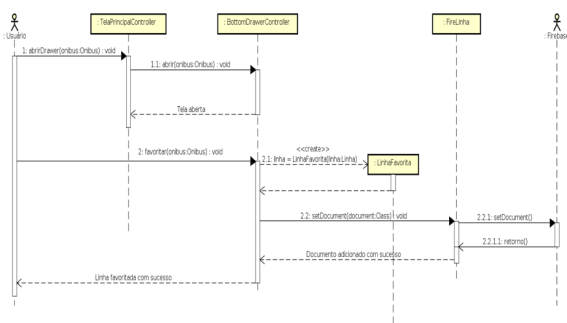


Fig. 5. Escolhendo uma linha de ônibus preferida

Uma vez definidas as funções e as operações do sistema, iniciou-se o processo de prototipagem, que define efetivamente como o usuário interage com o sistema para realizar suas ações e, conseqüentemente como o sistema responde ao usuário por estas ações. Desta forma, a concepção da interface precisa dar ao usuário acesso as funcionalidades do sistema de forma, rápida e intuitiva. Essa

etapa é importante também, para confirmar se as seqüências de ações definidas no caso de uso estão desenhadas de modo a atender os objetos definidos (Figura 6).

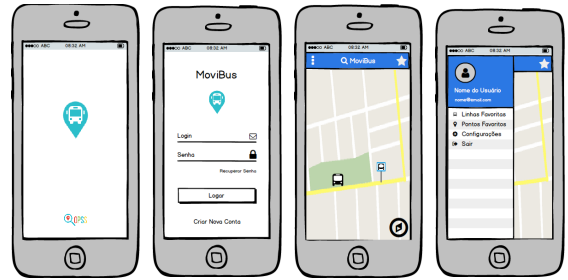


Fig. 6. Mockups: Telas de splash, login, principal e menu lateral

IV. RESULTADOS E DISCUSSÕES

Os três principais componentes do RITP (Tracker App, Movibus App e RITPBase) mostrados na Figura 1 foram implementados com suas funcionalidades essenciais. Com os módulos disponíveis, iniciou-se uma avaliação preliminar do funcionamento do sistema, onde duas rotas de ônibus foram mapeadas: as linhas R09 que faz o percurso UESB -Centro- UESB e a linha P52 que faz o percurso ConquistaVI-UESB-ConquistaVI. Os itinerários podem ser vistos na Figura 7, os arquivos de indexação gerados em formato JSON são exibidos na Figura 8 e o mapeamento realizado pelo Tracker é mostrado na Figura 9.

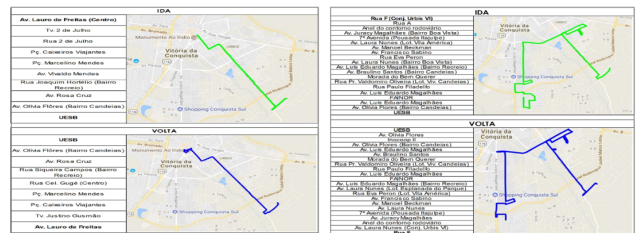


Fig. 7. Itinerários das linhas R09 (ida/volta) e P52 (ida/volta)

```

"name" : "uesb-centro-r09",
"pointLocationsArrayList" : [ {
  "latitude" : -14.89101906772703,
  "longitude" : -40.80399127677083
}, {
  "latitude" : -14.890037714503706,
  "longitude" : -40.804646238684654
}, {
  "latitude" : -14.891629563644528,
  "longitude" : -40.80606009811163
}, {
  "latitude" : -14.8925462923944,
  "longitude" : -40.80703491345048
}, {
  "latitude" : -14.8893428966403,
  "longitude" : -40.80721428617835
}, {
  "latitude" : -14.886835534125566,
  "longitude" : -40.80881221219897
}, {
  "latitude" : -14.87901249434799,
  "longitude" : -40.814175540581346
}, {
  "latitude" : -14.852290484122932,
  "longitude" : -40.837960951030254
}, {
  "latitude" : -14.853013674728572,
  "longitude" : -40.84099670872092
}
],
...
"name" : "uesb-conquistaiv-p52",
"pointLocationsArrayList" : [ {
  "latitude" : -14.89043447189033,
  "longitude" : -40.803696820512414
}, {
  "latitude" : -14.892562595196068,
  "longitude" : -40.80706567503512
}, {
  "latitude" : -14.889426464214921,
  "longitude" : -40.807114373892546
}, {
  "latitude" : -14.886856824159622,
  "longitude" : -40.8087848033756
}, {
  "latitude" : -14.879035670310259,
  "longitude" : -40.814155004918575
}, {
  "latitude" : -14.876332296989858,
  "longitude" : -40.816009836271405
}, {
  "latitude" : -14.873688980005682,
  "longitude" : -40.81779099069536
}, {
  "latitude" : -14.905602321960032,
  "longitude" : -40.840460266917944
}, {
  "latitude" : -14.906826498918235,
  "longitude" : -40.84045900963247
}
]
    
```

Fig. 8. Informações geográficas em JSON

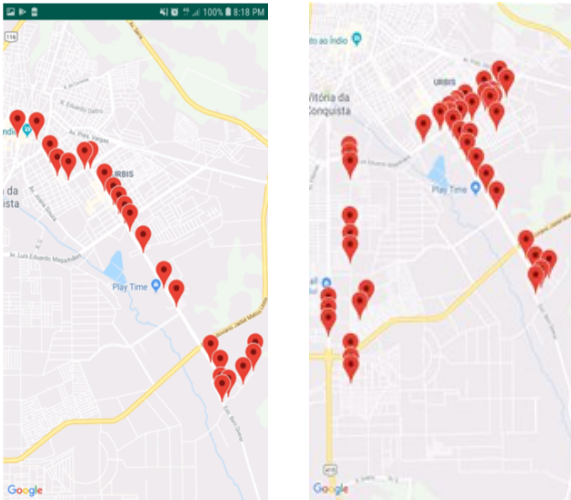


Fig. 9. Tracker: Mapeamento das linhas R09 e P52

Com estas informações disponíveis no RITP, o usuário pode utilizar o aplicativo *MoviBus* para interagir com o SPTU através das funcionalidades fornecidas por sua interface gráfica. O aplicativo permite o cadastro manual do usuário ou a utilização do seu *login* do Google (Figura 10a e 10b). Este cadastro permite o gerenciamento de itens como ônibus, paradas ou rotas favoritas facilitando o acesso aos dados mais utilizados. Estas opções podem ser acessadas através do menu lateral (Figura 10c) que permite visualizar e alterar os dados da conta, como a imagem de perfil, nome e e-mail. Abaixo tem-se acesso as Linhas Favoritas, Pontos Favoritos, Configurações e Sair que permitem ao usuário listar e configurar os itens correspondentes, além de efetuar o *logout* do aplicativo.

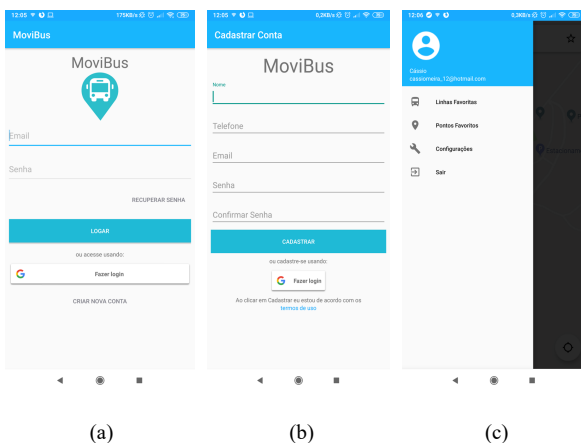


Fig. 10. MoviBus GUI: Telas de login, cadastro e menu lateral

Na parte superior da tela principal encontra-se o ícone de menu lateral à esquerda, um campo de pesquisa ao centro e um ícone de favoritos à direita. O campo de pesquisa permite ao usuário encontrar qualquer informação fornecida pelo sistema e o ícone de favoritos permite o acesso rápido a tela de favoritos, que mostra todos as linhas e pontos de

ônibus preferidos do usuário. A porção central da tela principal mostra um mapa da região onde o usuário se encontra geograficamente, nela é possível visualizar os *markets* que representam os objetos gráficos inseridos no mapa como, por exemplo, os ônibus e suas paradas (Figura 11a).

Quando o usuário clica sobre um *market*, um menu do tipo Bottom Drawer fornece informações sobre este elemento. No caso de um ônibus, sua origem, destino e itinerário são visualizados, com a possibilidade de escolher este ônibus como favorito. Além disso, também é possível realizar duas ações neste menu. A primeira ativa o acompanhamento de um ônibus em tempo real posicionando o *market* sempre no centro do mapa, à medida que o ônibus altera sua localização. A segunda visualiza o itinerário do ônibus permitindo ao usuário conhecer todo o percurso da linha e auxiliando em uma possível tomada de decisão (Figura 11b).

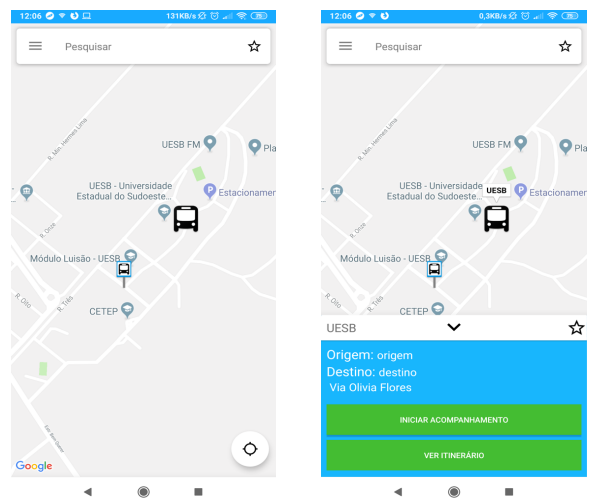


Fig. 11. MoviBus GUI: Tela principal e menu Bottom Drawer

A Figura 12 mostra uma avaliação preliminar da cobertura do sinal de dados nas quatro principais operadoras de telefonia celular da cidade de Vitória da Conquista - BA. O teste foi realizado em cinco viagens na Linha R09 com duração média de 18 minutos e 26 segundos. Mensagens de localização foram enviadas a cada 1 segundo, desta forma, supondo uma velocidade média de 60 Km/h, o sistema atualiza a posição do ônibus a cada 16,6 metros. Conforme visto na Figura 7, o itinerário desta linha ocorre na região central da cidade, que é bem servida de estações rádio base (ERBs). O resultado foi bastante satisfatório nas Operadoras 1 e 3. Contudo, notou-se uma falha na transmissão de dados nas Operadoras 2 e 4 em pequenos trechos do percurso. Em média, foram geradas 1.106 mensagens de atualização em cada conjunto de teste. As perdas variaram entre 0,83%, 4,59%, 7,52% e 9,17%.

As falhas ocorreram em pontos semelhantes, possivelmente por estas operadoras compartilharem as mesmas ERBs. Estas falhas pontuais reforçam a necessidade de uma metodologia de avaliação da Qualidade de Serviço (QoS) do sistema que está prevista para a próxima fase.

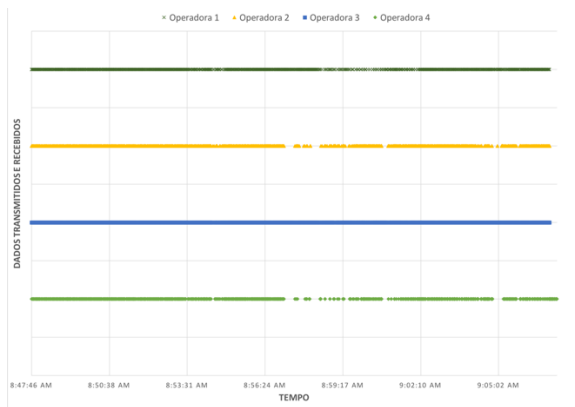


Fig. 12. Cobertura do sinal de dados

Embora plenamente funcional em sua atualidade, aspectos relacionados a dificuldade de testes, manutenção e implementação de um sistema deste porte com uma equipe pequena e volátil, bem como as restrições relacionadas ao armazenamento e a quantidade de acessos simultâneos permitidos na versão gratuita do Firebase [24], levaram a diminuição do escopo do problema a ser solucionado, passando do sistema de transporte público da cidade de Vitória da Conquista - BA para o sistema de transporte de funcionários da UESB, campus de Vitória da Conquista - BA. Essa nova estratégia permitirá ao grupo de pesquisa resolver os mesmos problemas da proposta original, contudo em um ambiente menor, mais controlado e amigável. Características como veículos, rotas, horários, entre outras, já foram implementadas no RITP (Figura 13). Desta forma, as próximas etapas das fases de integração, testes e refinamento do sistema serão realizadas neste novo *test bed*, permitindo futuramente que o projeto seja implementado no escopo original com mais facilidade.

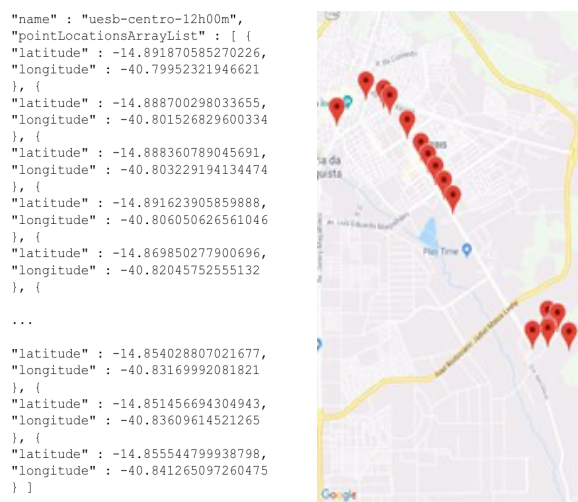


Fig. 13. Mapeamento do transporte de funcionários da UESB

V. CONCLUSÕES E TRABALHOS FUTUROS

O projeto RITP - Redes Inteligentes de Transporte Público desenvolvido pelo grupo de pesquisa OPSS - Otimização, Programação e Simulação de Sistemas possui

uma arquitetura complexa e está sendo desenvolvido em etapas. Neste contexto os módulos Tracker App, Movibus App e RITPBase foram implementados e disponibilizados, permitindo que o sistema, mesmo que experimentalmente, fosse utilizado satisfatoriamente pelos usuários. As próximas etapas consistem nas fases de integração, testes e refinamento do sistema, contudo devido as dificuldades relacionadas ao desenvolvimento acadêmico de um sistema deste porte, um novo *test bed* foi definido: o sistema de transporte de funcionários da UESB campus de Vitória da Conquista - BA, cujas características já estão mapeadas no RITP. Na sequência, estão previstos testes relacionados a QoS da cobertura do sinal de dados e cobertura do sinal de GPS, além funcionalidades de ETA.

AGRADECIMENTOS

Os autores agradecem a UESB e ao CNPq pelo apoio na forma de bolsas de Iniciação Científica e a todos os alunos voluntários que participaram do projeto, em especial ao discente João Ribeiro Andreotti pela valiosa colaboração na coleta de dados.

REFERÊNCIAS

- [1] Cramer, J., Krueger, A. B. (2016). Disruptive Change in the Taxi Business: The Case of Uber. *American Economic Review*, Vol. 106, No. 5, pp. 177-182.
- [2] Kawamoto, E. (2015). *Análise de Sistemas de Transporte*. EESC-USP, São Carlos, 229p.
- [3] ANTP (2018). *Relatório 2018 do Sistema de Informações da Mobilidade Urbana da Associação Nacional de Transporte Público*. São Paulo-SP.
- [4] Citymapper (2021). <<https://citymapper.com>>. Acessado em 03/05/2021.
- [5] Urbanoide. (2021). *Urbanoide app* <<http://www.urbanoide.com.br/>>. Acessado em 03/05/2021.
- [6] Cadê o ônibus? (2021). <<http://www.cadeonibus.com.br/>>. Acessado em 03/05/2021.
- [7] Vá de Ônibus. (2021). *Vá de Ônibus app* <<https://www.vadeonibus.com.br/>>. Acessado em 03/05/2021.
- [8] Moovit. (2021). *Moovit app*. <<https://moovitapp.com>>. Acessado em 03/05/2021.
- [9] CittaMobi (2021). *CittaMobi app*. <<https://www.cittamobi.com.br/home/onde-estamos/#collapse7892>>. Acessado em 03/05/2021.
- [10] Davis, I. (2020). What Are the Benefits of MVC? *Internet Alchemy*. Acessado em 25/11/2020.
- [11] Fowler, M. (2013). *UML Distilled: A Brief Guide to the Standard Object Modeling Language*, 3rd Edition, Addison-Wesley Professional, 208p.
- [12] Astah (2020). *A UML editor integrated with Mind Mapping features*. Disponível em: <<https://astah.net/editions/community>>. Acessado em 02/12/2020.
- [13] Balsamiq (2018). *Balsamiq Mockups*. Disponível em: <<https://balsamiq.com/products/Mockups/>>. Acessado em 10/12/2018.
- [14] Gartner (2019). *Gartner Says Global Smartphone Sales Stalled in the Fourth Quarter of 2018*. Disponível em <<https://www.gartner.com/en/newsroom/press-releases/2019-02-21-gartner-says-global-smartphone-sales-stalled-in-the-fourth-quarter>>. Acessado em 21/01/2019.
- [15] Smith, Craig. (2019). *Android statistics*. Disponível em <<https://expandedramblings.com/index.php/android-statistics/>>. Acessado em 10/06/2019.
- [16] Elprocus (2021). *What Everybody Ought to Know About Android: Introduction, Features & Applications*. Disponível em

- <<https://www.elprocus.com/what-is-android-introduction-features-applications/>>. Acessado em 30/07/2021.
- [17] Lecheta, R. (2015). Google Android. 5ª Edição, Novatec Editora, 1072p.
- [18] Umezu, C. K., Cappelli, N. L. (2002). Sistema de Baixo Custo para Geração de Sinal de Correção Diferencial, em Tempo Real, para GPS. XXXI Congresso Brasileiro de Engenharia Agrícola - CONBEA 2002, Salvador-BA.
- [19] Sant'anna, L.; Machado, R.; Brito, M. (2015). A logística reversa de resíduos eletroeletrônicos no Brasil e no mundo: o desafio da desarticulação dos atores. Sustentabilidade em Debate, v.6, n. 88.
- [20] Rossini, V; Sanches, S. H. D. F. N. (2017). Obsolescência programada e meio ambiente: A geração de resíduos de equipamentos eletroeletrônicos. Revista de Direito e Sustentabilidade, v.3, n.1, pp.51-71.
- [21] PNUMA. (2021). Programa das Nações Unidas para Meio Ambiente. Disponível em <<http://https://www.unenvironment.org>>. Acessado em 20/08/2021.
- [22] Firebase Google (2021a). Disponível em <<https://firebase.google.com/>>. Acessado em 14/08/2021.
- [23] Movibus. (2021). Movibus on Play Store. Disponível em: <<https://play.google.com/store/apps/details?id=com.opss.movibus>>. Acessado em 17/08/2021.
- [24] Firebase Google (2021b). Disponível em <<https://firebase.google.com/docs/firestore/quotas/>>. Acessado em 17/08/2021.