

Aplicativo de Geolocalização de Ônibus – Path Bus

Adenor Mário Ramos da Silva, Rubens Conceição da Silva Junior, Romulo Gabriel Rodrigues, Johelden C. Bezerra
Estácio – IESAM, Avenida Gov. José Malcher, 1148 - Nazaré - Belém-PA

Resumo — Este trabalho tem como foco a mobilidade urbana, a qual é um dos principais problemas das grandes cidades, e propõe o desenvolvimento do aplicativo Path Bus. Esse aplicativo será uma rede colaborativa que servirá para que o usuário dos transportes coletivos públicos tenham um meio de monitorar a posição aproximada de um ônibus, com isso poder planejar o tempo de sua viagem com mais precisão. O aplicativo terá uma base de conhecimento que será alimentada com dados que os próprios usuários fornecerão automaticamente por meio do GPS (Global Positioning System) do smartphone, no qual o aplicativo estará instalado.

Palavras-chaves — Crowdsourcing, aplicativo móvel, aplicativo multiplataforma, geolocalizador de ônibus, redes colaborativas.

1. INTRODUÇÃO

O acelerado crescimento desordenado das cidades e o aumento do índice populacional criam uma alta demanda de veículos de transportes públicos para atender a necessidade de deslocamento em massa de pessoas, no entanto a gestão pública juntamente com as empresas que são responsáveis por prover esses recursos, não conseguem atender a essa necessidade, o que causa um desgaste excessivo das frotas que já existem e contribui para a redução da qualidade de vida dos indivíduos dessa cidade.[1] Com a baixa qualidade dos serviços prestados e dos ônibus utilizados, isso pode ter contribuído para a diminuição na quantidade de passageiros, segundo a revista (NTUurbano), o ano de (2015) registrou uma queda de 9% na demanda de usuários pagantes se comparado a 2014, essa redução é três vezes maior do que foi registrado nos cinco anos anteriores. Essa diminuição de usuários deixou um grande prejuízo para as empresas que continuam fornecendo o mesmo serviço. Esse quadro pode ser revertido se houver investimento por parte do governo, em medidas para estimular o uso dos transportes coletivos e equilibrar o uso dos espaços públicos nas vias urbanas, já que 60% do espaço das vias, são utilizados por veículos particulares que transportam apenas 20% população, enquanto os ônibus ocupam 25% e transportam 70% das pessoas.[2] Em Belém, no ano de 2015 existiam 218.981 automóveis particulares enquanto que o número de ônibus era 3.806, a diferença é notável levando em consideração que uma quantidade tão pequena de ônibus são responsáveis por transportar a maior parte da população.[3]

A defasagem estrutural das cidades devido ao crescimento desordenado torna ineficaz o investimento no aumento da quantidade da frota de ônibus, pois o problema está na estrutura das ruas e avenidas das cidades, que não possuem capacidade para suportar um grande fluxo de veículos, sendo assim, uma medida para minimizar este problema seria o auxílio da tecnologia, onde a própria população irá gerenciar com transparência, por meio de aplicativos que façam a integração desse serviço em uma plataforma móvel, o emprego dessas tecnologias é determinante para a experiência dos usuários pois, o mesmo poderá planejar melhor uma rota de deslocamento para economizar tempo. [3]

A internet possibilita inúmeras formas de resolução de problemas, já que reduziu a fronteira que interliga as

pessoas, cada indivíduo com acesso à internet é um colaborador em potencial. Um sistema colaborativo, *crowdsourcing*, é uma forma de inteligência distribuída em que a informação inserida é constantemente alterada e em tempo real, que tem como base a premissa de que ninguém sabe tudo, todos sabem alguma coisa e todo o conhecimento reside na humanidade. [4]

As redes colaborativas são uma via de mão dupla onde cada usuário é um formador e consumidor de recursos, e todo conhecimento imputado por cada usuário é válido. Existem inúmeros exemplos de serviços ou recursos que só são mantidos porque a população mundial, ou local dependendo do contexto, gera e consome informações o tempo inteiro, como por exemplo, a Wikipédia, onde estão disponíveis informações dos mais variados assuntos, e em caso de um determinado tema não possuir qualquer informação, o usuário pode adicionar o seu conhecimento sobre o assunto e esse conhecimento inserido estará disponível para o resto do mundo consumir e modificar de acordo com as regras estabelecidas pelos moderadores do site, outro exemplo, bem interessante é o sistema operacional *open-source* GNU/Linux, onde os usuários contribuem para manter o sistema funcionando adequadamente, e todos trabalham juntos para corrigir falhas e melhorar o sistema para que o mesmo não deixe de existir. O Firemesh, usado na Austrália, é um aplicativo para reportar focos de incêndio, onde a população fica encarregado de inserir as informações da localização dos incidentes. [4]

Este trabalho é dividido nas seguintes sessões: - A sessão 2 apresenta a metodologia usada no desenvolvimento desse trabalho. Na sessão 3 são apresentadas as tecnologias utilizadas no desenvolvimento do aplicativo. Na sessão 4 encontra-se a descrição do desenvolvimento do aplicativo. Na sessão 5 encontra-se a conclusão do trabalho.

2. METODOLOGIA

A metodologia teve o início na observação do problema, para que fosse possível pensar em soluções tecnológicas que seriam viáveis, e que não acarretassem custos altos para a etapa de desenvolvimento, foi decidido que a melhor abordagem seria desenvolver um aplicativo em que o próprio usuário monitorasse a localização dos ônibus, no passo seguinte foi decidido o uso de tecnologias híbridas para que o aplicativo pudesse atingir o maior número de usuários possível com uma menor curva de aprendizado e menor complexidade de desenvolvimento, o embasamento teórico para o desenvolvimento do aplicativo foi obtido através de pesquisas em livros, fóruns de programação, acesso aos vídeos do youtube relacionados ao desenvolvimento de aplicativos híbridos, busca de cursos online relacionados ao desenvolvimento de aplicativos híbridos com o framework Ionic e o framework Angular.js, busca de materiais de linguagens web que são um dos pilares no desenvolvimento de aplicações multiplataforma, essas linguagens são o HTML, CSS e Javascript.

Para ter uma visão geral sobre os resultados e a tecnologia, foi necessário adquirir conhecimento mínimo para começar o desenvolvimento do aplicativo e definir os limites do escopo do desenvolvimento, a primeira pesquisa foi relacionada a *crowdsourcing* que é a forma como um serviço ou aplicativo adquire dados, para então gerar informações que serão fornecidas aos usuários. Como um dos objetivos desse trabalho é atingir ao maior número de usuários possível optou-se por desenvolver aplicativos multiplataforma, que reduz o custo de desenvolvimento, já que não é necessário desenvolver um código para cada plataforma. Para que o aplicativo funcione corretamente sem problemas de desempenho em relação ao serviço oferecido, é necessário ter um bom sistema de distribuição, logo é importante ter conhecimento das técnicas utilizadas em sistemas distribuídos, para que pudessem ser aplicadas no servidor do aplicativo.

O estudo das tecnologias foi fundamental na etapa de desenvolvimento, pois sem as bases da arquitetura REST (*Representational State Transfer*), que são um conjunto de regras no desenvolvimento de sistemas distribuídos, onde o foco central é o recurso, e algumas dessas regras são uso de modelo cliente servidor, interação sem estado e interface uniforme.

Node.js, que é um ambiente de execução de javascript, foi usado na construção da API de serviço, onde estão todas as regras de negócio. A plataforma Heroku foi escolhida para hospedar o serviço por ser fácil de utilizar e gerenciar toda a aplicação, como por exemplo, se existir uma grande demanda de usuário, o Heroku aumenta a quantidade de máquinas para atender aos usuários, quando número de requisições de usuários diminuírem, ele reduz automaticamente, dando elasticidade ao serviço. A etapa de implementação teve início com o desenvolvimento paralelo do aplicativo de usuário e API, onde foi possível desenvolver e realizar pequenos testes ao longo do processo, a etapa de correção de erros iniciou-se após os testes que identificarão os problemas.

Por último a construção do artigo referente a todo o processo realizado no decorrer do desenvolvimento e a revisão bibliográfica de todo o conteúdo utilizado nas etapas descritas anteriormente, conforme a Fig. 1.

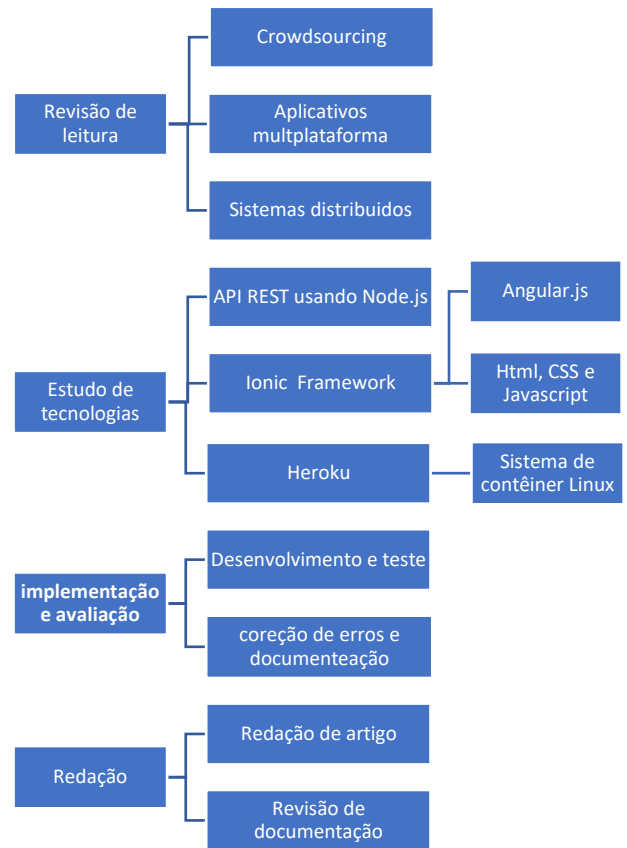


Fig. 1 – Diagrama de blocos da metodologia.

3. TECNOLOGIAS UTILIZADAS

3.1 Ionic Framework

Ionic é um plataforma de desenvolvimento de software gratuito, que fornece uma estrutura prática, acessível e fácil, para o desenvolvimento de aplicações multiplataforma, também utiliza o padrão de desenvolvimento MVC (modelo, visão e controle). A utilização do Ionic, nesse projeto, deve-se a pequena curva de aprendizagem, pois utiliza tecnologias web como HTML5, CSS e javascript que são responsáveis pela estrutura, estilização e ações do aplicativo. [5]

3.2 Apache Cordova

Tecnologia que faz a integração do aplicativo, que normalmente não possui acesso ao hardware por se tratar de uma aplicação web, aos recursos dos diferentes dispositivos móveis, possibilitando que o aplicativo tenha acesso aos recursos do aparelho no qual está instalado, como por exemplo o GPS, câmera, etc. [6]

3.3 Angular.js

Framework que usa linguagem javascript, open-source mantido pela google, que é responsável em construir e executar SPA (single-page-application). Um de seus objetivos é adicionar funcionalidades extras ao HTML, já que utiliza as estruturas da mesma, identificando as tags marcadas através de suas diretivas, e realizando ações nas mesmas. O Angular.Js melhora a experiência com conteúdo dinâmico devido a ligação direta e bidirecional de dados (two-way data-binding), que faz a sincronização imediata entre models e views, isso faz com que a aplicação seja mais rápida pois, o carregamento da view é feito apenas uma única vez, modificando o conteúdo que é exibido nas páginas de forma dinâmica. [7]

3.4 Node.js

Interpretador de código javascript que funciona no servidor, sendo fácil de implementar, rápido e escalável para construir aplicações de rede. Possui o maior ecossistema de pacotes open-source do mundo chamado NPM (Node Package Manager), essa tecnologia foi utilizada na construção da API (Application Programming Interface) que é responsável por manter o serviço funcionando corretamente, é a API quem vai receber os dados de localização enviados pelo usuário, tratar e disponibilizar informações da localização dos ônibus. [8]

3.5 Heroku

Plataforma de serviço em nuvem que suporta várias linguagens de programação como java, Node.js, Scala, Clojure, Python entre outras.[9]

3.6 Android SDK

É um kit de desenvolvimento para a plataforma Android que fornece ferramentas ao desenvolvimento de aplicativos para todas as versões dos sistemas operacionais de celulares android existentes no mercado, ele é necessário para realizar testes virtuais, pois com ele é possível emular um sistema Android no computador utilizado no desenvolvimento ou testar nativamente em um aparelho *smartphone*, para isso, faz-se necessário ter instalado no Android SDK a versão do sistema Android na qual se deseja realizar o teste e alguns outros pacotes recomendados, como por exemplo o ADB (Android Debug Bridge). [10]

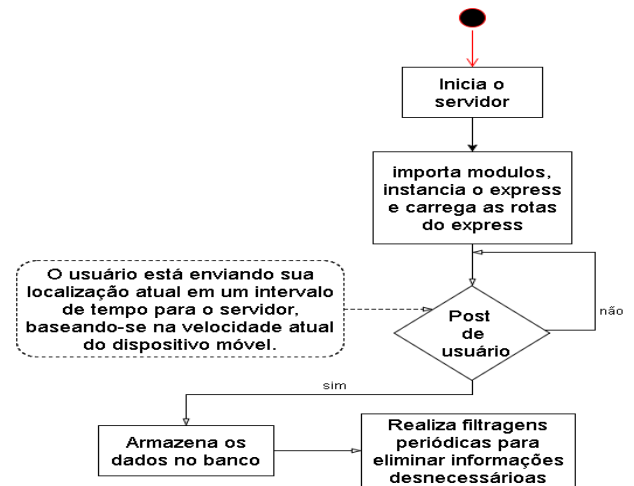
4. DESENVOLVIMENTO DO APLICATIVO

Por ser uma aplicação que irá rodar no aparelho *smartphone* e disponibilizar informações obtidas do tratamento dos dados recebidos de cada usuário, precisou ser desenvolvido em duas partes, a aplicação que roda no *smartphone*, que será responsável por exibir as informações ao usuário, monitorar sua localização e enviar ao servidor caso seja verificado que o usuário está a uma velocidade sobre-humana, e a aplicação que roda no servidor, a API que será responsável por prover as informações solicitadas pelo usuário.

4.1 API PBServer

O servidor que gerencia os recursos, foi desenvolvido com a plataforma Node.js. O servidor conta com um método que recebe os dados de geolocalização enviados pelo usuário que está em um ônibus, e armazena no banco de

dados SQL para garantir a integridade dos dados, com



forme a Fig. 2.

Fig. 2 – Funcionamento da recepção de dados do usuário.

Uma resposta será enviada ao usuário com a posição do ônibus selecionado, baseado nos dados coletados de cada usuário, que está mais próximo, contendo os parâmetros enviados junto com a requisição, é realizada uma filtragem nos dados do servidor, para então, responder ao usuário com a informação solicitada, conforme a Fig. 3.

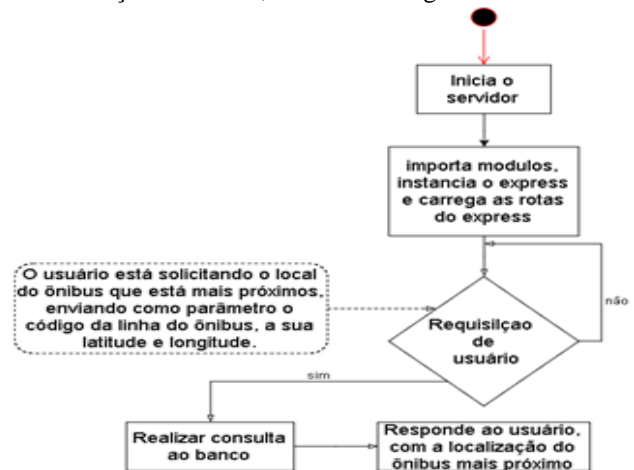


Fig. 3 – Funcionamento da requisição do usuário.

Para que o desenvolvimento da API fosse mais rápido, foi utilizado juntamente com o Node.js, o framework Express.js, e para que a API pudesse ser mais versátil ao trabalhar com a ORM sequelize que é altamente acoplável ao javascript, para facilitar as operações sobre banco de dados SQL.

A implementação da API foi realizada na plataforma Heroku que disponibiliza um sistema de containers em máquinas virtuais com sistema operacional Linux.

4.2 Aplicativo Path Bus

O aplicativo foi desenvolvido utilizando a tecnologia para construção de aplicações multiplataforma Ionic, que disponibiliza vários recursos de estilização e estrutura como campos de texto, botões e cards estilizados que facilitam e

ajudam a acelerar o trabalho do desenvolvedor, além disso, toda a estrutura básica para se começar um projeto como, por exemplo, as pastas onde ficam localizados os arquivos de plugins, HTML, CSS e Javascript, são gerados automaticamente no momento da criação do projeto através de comandos inseridos no terminal, comandos que são informados na instrução de instalação e configurações iniciais disponível no site.

O desenvolvimento de aplicativos com Ionic é feito de forma similar ao desenvolvimento de um site web, a estrutura de páginas e diretórios é bem parecida com o que é feito no desenvolvimento de um site, no entanto, há uma diferença que faz com que o aplicativo pareça estar rodando nativamente, isso deve-se a forma de como o Ionic utiliza a página do navegador, ou *view*, a página é reorganizada e todos os componentes como a barra de endereço, barra de títulos e todos os outros componentes padrões são removidos restando apenas uma página em branco, onde o Ionic renderia as informações.

O aplicativo desenvolvido é um protótipo que possui as funcionalidades necessárias para a prestação do serviço proposto, para isso o aplicativo conta com uma tela principal na qual é realizada uma animação no instante em que o aplicativo é iniciado, essa animação inicia no centro do mapa e com zero de zoom, enquanto o GPS não encontra a latitude e longitude, fica sendo mostrado na tela uma mensagem que indica que o aplicativo está buscando a localização, no momento em que o GPS encontra o local, é disparada uma animação que aproxima a câmera até a posição encontrada, em um tempo de 4 segundos, onde será visualizado um marcador no centro da tela, o mapa utilizado foi o da google e a animação foi implementada utilizando o *plugin* cordova-plugin-googlemaps que possui código aberto e é fornecido por Masashi Katsumata através do Github, nessa página principal, existem botões localizados na parte inferior da tela, que são botões de navegação, um que serve para o usuário voltar para a tela inicial do aplicativo, outro para a tela de lista de ônibus e um para a tela de login, conforme a Fig 4.

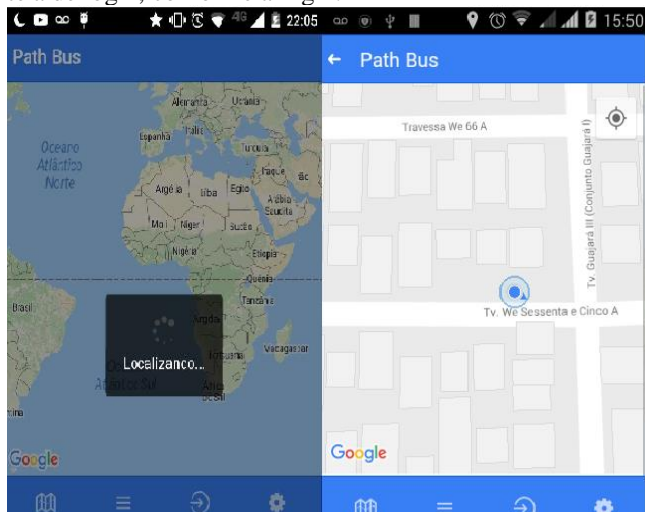


Fig. 4 – Tela de inicialização e home do smartphone.

A tela de listar que será exibida após o usuário pressionar o botão, exibirá uma lista com o nome das linhas dos ônibus da cidade de Belém no lado esquerdo da tela, já no lado direito será exibido o código referente ao ônibus listado, o usuário poderá percorrer a lista para encontrar o ônibus desejado, ou poderá clicar no campo de busca localizado a cima da lista, possibilitando ao usuário, filtrar pelo nome ou pelo código da linha, conforme a Fig 5.



Fig. 5 – Tela de listar do smartphone.

Após o usuário encontrar o nome do ônibus desejado na lista, ele poderá clicar e o aplicativo irá enviar uma solicitação ao servidor informando a posição atual do usuário e o código do ônibus desejado, o servidor devolverá as informações solicitadas pelo aplicativo e o mesmo encaminhará o usuário para a tela inicial onde será mostrado no mapa um marcador indicando o local aproximado do ônibus, conforme a Fig 6.



Fig. 6 – Tela mostrando a localização do ônibus.

O aplicativo também possui uma tela de login, abertura de conta e recuperação de senha, no entanto é preciso deixar

bem claro que o aplicativo não irá relacionar o usuário com a informação enviada para o servidor para evitar que o usuário seja rastreado, as informações do usuário servirão apenas para criação de conta e acesso ao sistema. A tela de login possui formato padrão com um campo para nome e outro para senha, após os campos de texto encontra-se o botão de enviar. Caso de o usuário ainda não tenha acesso ao aplicativo, na tela de login existe um botão para criação de conta, por outro lado, se o usuário já cadastrado esqueceu sua senha, o mesmo pode criar uma conta nova ou recuperar a senha de uma conta existente através dos botões localizados abaixo do botão de enviar, conforme a Fig 7.



Fig. 7 – Tela de login, cadastro e recuperação de senha.

5. CONCLUSÃO

Com o desenvolvimento do aplicativo proposto, conclui-se que é possível otimizar o tempo de esperas nas paradas de ônibus por meio de tecnologias que auxiliem ao usuário, ajudando-os a localizar o veículo que pretende usar em seu deslocamento.

O aplicativo está pronto para uso em uma versão beta, que foi testada em smartphones com sistema operacional android, nas versões 4.4.2, 5.0 e 6.0, no entanto para a plataformas IOS não foram realizados testes no hardware por não ter acesso ao mesmo e sim em uma virtualização gerada pelo próprio Ionic através do comando *ionic lab* que abre uma tela que mostra a virtualização do aplicativo nas plataformas instalados no Ionic, no entanto como o aplicativo está funcionando no android, deve funcionar sem problemas em sistemas iOS.

REFERÊNCIAS

- [1] C. Magagnin, Renata; N. R. Silva, Antônio, A percepção do especialista sobre o tema mobilidade urbana, *Transportes*, 25–35, 2008.
- [2] “NTUurbano,” vol. 36, pp. 101–105, 2014.
- [3] “IBGE | Cidades | Pará | Belém | Frota - 2015.,” <http://www.cidades.ibge.gov.br/xtras/temas.php?la>

- [4] D. Rafael, Lucio, “Um aplicativo para dispositivos móveis voltado para usuários de transporte público, ”, UTRPR, paraná, 2011.
- [5] Ionic framework. Disponível em: <http://ionicframework.com/>. Acessado em: 28/11/2016.
- [6] Apache Cordova. Disponível em:
- [7] Angular.js. Disponível em: <https://angularjs.org/>. Acessado em: 28/11/2016.
- [8] Node.js. Disponível em: <https://nodejs.org/en/>. Acessado em: 28/11/2016.
- [9] Heroku. Disponível em: <https://www.heroku.com/>. Acessado em: 28/11/2016.
- [10] Android SDK. Disponível em: <https://developer.android.com/studio/intro/update.html?hl=pt-br>. Acessado em: 28/11/2016.
- [11] Robson Sales, “IBGE: Mais de 50% usam celular e tablet para acessar a internet | Valor Econômico.,” Disponível em: <http://www.valor.com.br/brasil/4027294/ibge-mais-de-50-usam-celular-e-tablet-para-acessar-internet>, “Acessado em: 05/10/2016”
- [12] Rafael Menezes, “Como os aplicativos móveis podem ajudar a sociedade civil? - Observatório do 3º Setor.,” Disponível em: <http://observatorio3setor.com.br/carrossel/aplicativos-moveis-podem-ajudar-sociedade-civil/>, “Acessado em:07/10/2016”.