

UNIVERSIDADE ESTADUAL DO MARANHÃO
CENTRO DE CIÊNCIAS TECNOLÓGICAS
CURSO DE ENGENHARIA DA COMPUTAÇÃO

MARCOS SOUSA DA PENHA

**ROTA DE RODAS: desenvolvimento de um aplicativo móvel para o auxílio de pessoas
com deficiência locomotora e mobilidade reduzida**

São Luís
2017

MARCOS SOUSA DA PENHA

ROTA DE RODAS: desenvolvimento de um aplicativo móvel para o auxílio de pessoas com deficiência locomotora

Trabalho de Graduação apresentado ao Curso de Engenharia de Computação da Universidade Estadual do Maranhão (UEMA), como requisito para obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Dr. REINALDO DE JESUS DA SILVA

São Luís
2017

Penha, Marcos Sousa da.

Rota de rodas: desenvolvimento de um aplicativo móvel para auxílio de pessoas com deficiência locomotiva / Marcos Sousa da Penha. – São Luís, 2017.

65 f.

Monografia (Graduação) – Curso de Engenharia de Computação, Universidade Estadual do Maranhão, 2017.

Orientador: Prof. Dr. Reinaldo de Jesus da Silva.

1. Aplicativo. 2. Acessibilidade. 3. Deficiência. 4. Locomoção.
5. Mobilidade. I. Título.

CDU 004.4056.26

MARCOS SOUSA DA PENHA

ROTA DE RODAS: desenvolvimento de um aplicativo móvel para o auxílio de pessoas com deficiência locomotora

Trabalho de Graduação apresentado ao Curso de Engenharia de Computação da Universidade Estadual do Maranhão (UEMA), como requisito para obtenção do grau de Bacharel em Engenharia de Computação.

Trabalho Aprovado. São Luís, MA, 27 de julho de 2017

Prof. Dr. Reinaldo de Jesus da Silva
Orientador

Profa. Mestre Marilda de Fátima Lopes Rosa
Primeiro membro

Prof. Esp. João Batista de Oliveira Paiva Júnior
Segundo membro

Dedicatória

Dedico este trabalho primeiramente a Deus, aos meus pais, meu irmão, minha noiva, todos os demais familiares e amigos por todo o apoio que me deram em toda essa jornada.

AGRADECIMENTOS

Primeiramente a Deus, por ter me concedido saúde para chegar até aqui.

Aos meus pais, Ivoneide Lisboa de Sousa e Antonio Hermogênio da Penha, por toda a dedicação, amor, carinho e apoio por todos esses anos, e por terem me fornecido todas as condições para alcançar mais esta etapa.

Ao meu irmão, Marcelo Sousa da Penha, por todo o apoio, incentivo ao longo de todos esses anos.

A minha noiva, Dagmara Jagiello, que me deu suporte, me ouviu e me confortou em todos os momentos mais difíceis.

Aos meus tios, Loide Lisboa e Ocior Lopes, por todo o cuidado e suporte ao longo da minha graduação.

Ao meu orientador, que dedicou parte de seu tempo para acompanhar esta pesquisa e por acrescentar bastante valor a mesma.

A todos os meus amigos, que me ajudaram e me deram suporte.

*“Tudo tem o seu tempo determinado, e há tempo
para todo o propósito debaixo do céu.”
(Eclesiastes capítulo 3, versículo 1)*

RESUMO

Hoje no Brasil, temos um grande número de pessoas com deficiência locomotora. E além dessa limitação causada por tal problema, essas pessoas precisam lidar com a falta de estrutura para que as mesmas possam transitar dentro das cidades com liberdade e independência, o acesso a alguns lugares pode se tornar um grande desafio para essas pessoas. Garantir a acessibilidade para pessoas com dificuldades de locomoção é de grande importância, por isso foi pensado em aplicativo móvel colaborativo onde as pessoas com deficiência possam encontrar informações de lugares, ruas, avenidas ou estabelecimentos que possuam boa ou má acessibilidade. Neste trabalho são abordados conceitos de deficiência, acessibilidade e mobilidade, juntamente com tecnologias e técnicas de modelagem visando o desenvolvimento do aplicativo “Rota de Rodas” para dispositivos móveis com o sistema operacional *Android*. Esse aplicativo tem como objetivo, ser uma ferramenta colaborativa para os usuários, onde os mesmos poderão avaliar locais segundo seu nível de acessibilidade para deficientes locomotores. Os locais são divididos em duas categorias: locais internos (prédios, estabelecimentos) e locais externos (ruas, avenidas, praças). Para cada categoria há um questionário que possui cinco opções como resposta: péssimo, ruim, regular, bom e ótimo.

Palavras-chave: Aplicativo. Acessibilidade. Colaborativo. Locomoção. Mobilidade. Deficiência.

ABSTRACT

People with disabilities constantly experience difficulties with moving in Brazilian cities due to their poor structure which does not encourage disabled inhabitants to commune independently. The mobile crowdsourcing application was created to help them to easily access the places without fear of moving without any assistance. The work focuses on the concepts of disability, accessibility and mobility using the development modeling techniques and technologies of the “Rota de Rotas” application, which was created for mobile devices with the Android operating system. The goal is to allow users to evaluate places by their level of accessibility using a crowdsourcing tool. The destinations are divided into two categories: inside places (closed places, for example buildings) and outside places (such as street avenues and squares). Each category will be evaluated by users who will mark it as terrible, bad, regular, good or great.

Keywords: Application. Accessibility. Crownsourcing. Locomotion. Mobility. Deficiency.

LISTA DE ILUSTRAÇÕES

Figura 1 – Os processos inerentes na atividade de locomoção adaptada.....	19
Figura 2 – Esquema básico do modelo MVC.....	24
Figura 3 – Troca de informação entre cliente e servidor com a tecnologia <i>REST</i>	27
Figura 4 – Ciclo de vida de um serviço <i>SOAP</i>	28
Figura 5 – Versão web do <i>Wheelmap</i>	30
Figura 6 – Versão mobile do <i>Wheelmap</i>	31
Figura 7 – Aplicativo Guiaderodas.....	32
Figura 8 – Telas do aplicativo <i>WheelMate</i>	33
Figura 9 – Exibição do mapa no aplicativo.....	35
Figura 10 – Diagrama de Caso de uso do sistema Rota de Rodas.....	37
Figura 11 – Diagrama de classes do sistema Rota de Rodas.....	38
Figura 12 – Tela de autenticação.....	40
Figura 13 – Tela de erro de autenticação.....	40
Figura 14 – Tela de cadastro do aplicativo.....	41
Figura 15 – Menus do aplicativo.....	42
Figura 16 – Formulário de novo local (área externa).....	43
Figura 17 – Formulário de novo local (área interna).....	44
Figura 18 – Mapa.....	45
Figura 19 – Mapa com detalhes do local.....	45
Figura 20 – Avaliação de um local.....	45
Figura 21 – Nova avaliação de um local.....	45
Figura 22 – Tela de perfil do usuário.....	46

LISTA DE TABELAS

Tabela 1 – Comparação entre os aplicativos

Tabela 2 – Funções *Eloquent* ORM

LISTA DE SIGLAS

IBGE – Instituto Brasileiro de Geografia e Estatística

UML – *Unified Modeling Language*

MEC – Ministério da Educação

ABNT – Associação Brasileira de Normas Técnicas

SDK – *Software Development Kit*

MVC – *Model View Controller*

HTML – *HyperText Markup Language*

CSS – *Cascading Style Sheets*

PHP – *HyperText Preprocessor*

HTTP – *Hypertext Transfer Protocol*

FTP – *File Transfer Protocol*

REST – *Representational State Transfer*

XML – *Extensible Markup Language*

JSON – *JavaScript Object Notation*

SOAP – *Simple Object Access Protocol*

SMTP – *Simple Mail Transfer Protocol*

WSDL – *Web Services Description Language*

UDDI – *Universal Description, Discovery and Integration*

API – *Application Programming Interface*

Sumário

1. INTRODUÇÃO	15
1.1 Contribuições	15
1.2 Objetivo Geral	16
1.3 Objetivos Específicos	16
1.4 Apresentação do Trabalho	16
2. FUNDAMENTAÇÃO TEÓRICA	18
2.1 Deficiência	18
2.2 Mobilidade e Acessibilidade	19
2.3 Aplicativos Colaborativos.....	20
2.4 Aplicativos Móveis.....	22
2.4.1 Desenvolvimento de aplicativos móveis	23
2.5 Padrão MVC.....	24
2.6 Frameworks.....	25
2.6.1 AngularJS	25
2.6.2 Ionic Framework.....	25
2.6.3 Laravel Framework.....	26
2.7 Linguagem de Programação PHP.....	26
2.8 Web Service.....	27
2.8.1 REST.....	27
2.8.2 SOAP.....	28
2.9 Google Maps	28
3. TRABALHOS RELACIONADOS	30
3.1 Wheelmap.....	30
3.2 Guiaderodas	31
3.3 WheelMate	32
3.4 Comparação dos aplicativos.....	33
4. ARQUITETURA DO SISTEMA	35
4.1 Rota de Rodas.....	35
4.2 Definição do Sistema	36
4.2.1 Plataforma	36
4.2.2 Framework Usados.....	36
4.3 Modelagem do Sistema	37
4.3.1 Caso de Uso	37
4.3.2 Diagrama de Classes.....	38
5. PROTOTIPAGEM DO APLICATIVO ROTA DE RODAS	40
5.1 Elementos da tela do aplicativo	40
5.1.1 Tela de autenticação do aplicativo.....	40
5.1.2 Tela de cadastro de novo usuário	41
5.1.3 Menu do aplicativo.....	41
5.1.3 Tela de novo local.....	42
5.1.4 Mapa	44
5.1.5 Perfil de Usuário.....	46
6. CONCLUSÃO	47
6.1 Trabalhos Futuros.....	47
REFERÊNCIAS	48
APÊNDICES	53
APÊNDICE A – CÓDIGO CONTROLLER.JS	53

1. INTRODUÇÃO

Apesar da crescente evolução tecnológica nas últimas décadas, ainda são notórias as dificuldades das Pessoas com Necessidades Especiais ao acesso à informação e interação com a sociedade. Os recursos de acessibilidade disponíveis em ambientes computacionais contribuem para a inclusão social desses indivíduos, bem como para aspectos educacionais. No Brasil há 45,6 milhões de habitantes com algum tipo de deficiência, desse total, 13,2 milhões possuem deficiência motora, segundo o censo 2010 realizado pelo IBGE (Instituto Brasileiro de Geografia e Estatística).

Na cidade de São Luís no estado do Maranhão há aproximadamente 250 mil pessoas com deficiência, 11 % delas possuem deficiência motora com grande dificuldade de locomoção. No entanto, a cidade possui um problema alarmante no que tange a acessibilidade para as pessoas com deficiência e/ou mobilidade reduzida. O acesso lugares como praças, banheiros públicos, ou até mesmo a locomoção nas ruas, podem se tornar grandes desafios para essas pessoas.

Garantir a acessibilidade para pessoas com dificuldades de locomoção é de grande importância, baseando-se neste contexto, foi proposto o desenvolvimento de um aplicativo móvel colaborativo onde as pessoas com deficiência possam encontrar informações de lugares, ruas, avenidas ou estabelecimentos que possuam boa ou má acessibilidade dentro da cidade de São Luís, capital do estado do Maranhão.

Com a popularização de aparelhos *smartphones* e o fácil acesso à internet, vemos o grande alcance que os aplicativos desse tipo podem ter. Com o uso massivo e efetivo dessa aplicação, é possível mostrar aos governantes, donos de estabelecimentos e a população em geral, a importância de garantir acessibilidade para todas as pessoas.

A metodologia adotada no primeiro momento, uma abordagem comparativa será tomada, a partir do levantamento de pesquisas na área de acessibilidade e tecnologia, com o objetivo de analisar suas abordagens e resultados com o objetivo proposto deste trabalho. Tais comparações serão feitas a partir de critérios estabelecidos de forma quantitativa.

Para Gerhardt e Silveira (2009), a pesquisa qualitativa não se interessa com números, mas, com a compreensão de um grupo social, de uma organização, etc. Os pesquisadores que decidem adotar a abordagem qualitativa opõem-se ao pressuposto que defende um único modelo de pesquisa para todas as ciências, uma vez que as ciências sociais têm sua especificidade, o que pressupõe uma metodologia própria.

A pesquisa se caracterizará como exploratória e experimental, com levantamento e análise documental sobre o tema, mapeamento da do campus Paulo VI da Universidade Estadual do Maranhão, com pouca ou nenhuma acessibilidade, análise de aplicativos existentes e testes feitos para validar o aplicativo. Com base nos pressupostos evidenciados, considera-se relevante desenvolver um aplicativo móvel colaborativo voltado para acessibilidade de pessoas com deficiência física. Neste trabalho, defende-se o desenvolvimento de um aplicativo móvel colaborativo, levando-se em consideração as áreas internas da Universidade Estadual do Maranhão – UEMA para pessoas com mobilidade reduzida.

1.1 Contribuições

As principais contribuições que se pretende com o desenvolvimento deste trabalho são:

- Apresentar um mecanismo para avaliar acessibilidade de pessoas com deficiência física no campus Paulo VI - UEMA.
- Utilizar aplicativo móvel colaborativo para acessibilidade;
- Facilitar a localização de espaço sem acessibilidade para pessoas com mobilidade;
- Disponibilizar aplicativo móvel colaborativo para acessibilidade de pessoas com deficiência física e/ou mobilidade reduzida;
- E, por fim, criar o protótipo de uma ferramenta que fomente as discussões sobre de espaço físico para pessoas com mobilidade.

Portanto, o desafio deste trabalho é a localização de espaço com acessibilidade de pessoa reduzida na Universidade Estadual do Maranhão. Neste sentido, propõe um sistema móvel colaborativo para acessibilidade de pessoas com mobilidade reduzida. Em seguida, abordaremos os objetivos do trabalho.

1.2 Objetivo Geral

O objetivo geral deste trabalho é propor um aplicativo móvel colaborativo voltado para acessibilidade de pessoas com deficiência física e/ou mobilidade reduzida em algumas áreas internas da Universidade Estadual do Maranhão – UEMA através de imagens, assim como suas respectivas coordenadas, onde seus usuários poderão contribuir com sugestões para implementação de melhorias.

1.3 Objetivos Específicos

Para que o objetivo geral possa ser atingido, são propostos os seguintes objetivos específicos:

- Identificar e analisar os fundamentos sobre as condições de locomoção para pessoas com deficiência motora dentro da cidade de São Luís;
- Realizar o levantamento bibliográfico dos principais modelo e técnicas de sistema de colaborativos;
- Apresentar arquitetura de sistema colaborativo móvel para pessoas mobilidade reduzida;
- Disponibilizar ao usuário de um dispositivo móvel acesso ao sistema de colaborativos para pessoas com mobilidade reduzida.

1.4 Apresentação do Trabalho

Esta pesquisa está dividida em oito capítulos, dispostos da seguinte maneira:

- O capítulo 1 aborda a introdução da pesquisa. Apontando a metodologia, os objetivos e a motivação que deu início ao desenvolvimento do projeto;
- O capítulo 2 trata da fundamentação teórica, que é a base do conhecimento do projeto. Nesse capítulo os conceitos de deficiência, acessibilidade e mobilidade, aplicativos colaborativos, frameworks e webservices são apresentados;
- O capítulo 3 buscar relacionar aplicativos que possuam os objetivos semelhantes, para uma comparação com o que esse trabalho propões;

- O capítulo 4 descreve a arquitetura do sistema. Nesse capítulo, todos os detalhes do sistema são apresentados, desde a plataforma escolhida as modelagens com a linguagem UML;
- O capítulo 5 aborda a prototipagem do aplicativo, exibindo todas as suas funcionalidades, assim como sua interface;
- O capítulo 6 conclui o trabalho, e trás as propostas de trabalhos futuros relacionados ao sistema.

2. FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os conteúdos que abrangem o projeto desta monografia. Eles proporcionam uma base sólida e um bom alicerce para sustentar a ideia que o projeto apresenta. Serão abordados alguns conceitos como o de deficiência, conceitos de mobilidade e acessibilidade seguindo pelos aplicativos colaborativos e aplicativos móveis.

2.1 Deficiência

Segundo a Lei nº 7.853, de 24 de outubro de 1989, artigo 3º, deficiência é “toda perda ou anormalidade de uma estrutura ou função psicológica, fisiológica ou anatômica que gere incapacidade para o desempenho de atividade, dentro do padrão considerado normal para o ser humano”.

Segundo Diniz (2007) deficiência não deveria ser entendida como um problema individual, mas sim uma questão social. A pessoa que nasceu ou adquiriu uma lesão (ausência parcial ou total de um membro ou mecanismo corporal defeituoso), possui o difícil acesso à sociedade, pois ela deveria expressar sua forma corporal de estar no mundo, considerando que essa é uma das várias possibilidades para a existência humana.

Pode - se definir a deficiência física como "diferentes condições motoras que acometem as pessoas comprometendo a mobilidade, a coordenação motora geral e da fala, em consequência de lesões neurológicas, neuromusculares, ortopédicas, ou más formações congênitas ou adquiridas" (MEC, 2006). As pessoas com deficiência física, são aquelas que apresentam alterações musculares, ortopédicas, articulares ou neurológicas que podem comprometer sua locomoção. A deficiência física pode ser:

- Temporária: pessoa que sofre do problema pode voltar à suas condições anteriores a partir de um tratamento adequado;
- Recuperável: quando uma melhora é possível mediante tratamento, ou suplência por outras áreas não atingidas;
- Definitiva: a pessoa que sofre com a deficiência não apresenta chances de cura ou melhora;
- Compensável: quando a pessoa com deficiência pode ter uma melhora mediante a substituição de órgãos, por exemplo, a amputação compensável pelo uso da prótese (MEC, 2006).

Pode ser ocasionada por fatores hereditários, quando resulta de doenças transmitidas por genes, podendo manifestar - se desde o nascimento, ou aparecer posteriormente, por fatores congênitos, quando existe no indivíduo ao nascer e, mais comumente, antes de nascer, isto é, durante a fase intrauterina, ou adquirida, quando ocorre depois do nascimento, em virtude de infecções, traumatismos, intoxicações.

A atividade de locomoção faz parte do dia a dia de toda e qualquer pessoa. A locomoção é o ato de mover-se de um local para outro (HERSH; JHONSON, 2008). No processo de locomoção estão envolvidas diversas etapas como: conhecer o caminho a ser seguido, evitar os obstáculos, mudar de direção, se necessário, reconhecer o local de chegada, realizar a atividade desejada, retornar para o local de origem. O processo de locomoção é dividido em duas categorias: mobilidade e acesso ao ambiente. Segue abaixo uma visão geral sobre as duas categorias. Processos inerentes aos processos de locomoção adaptada.

Figura 1 - Os processos inerentes na atividade de locomoção adaptada.



Fonte: (HERSH; JHONSON, 2008)

A pesquisa em questão se concentra em dois aspectos principais: categoria e tarefas de mobilidade. Para um deficiente físico, conhecer o caminho é essencial, assim como a habilidade de evitar os obstáculos pelo caminho, no entanto, o meio em que ele se encontra precisa estar apto (adaptado) para isso. Para um deficiente físico, exercer tais tarefas se torna arduamente difícil, devido às suas limitações, tanto quando a falta de acessibilidade disponível. O aplicativo que está sendo desenvolvido, visa colaborar para minimizar esses problemas exibindo o melhor caminho, com o menor número de obstáculos possíveis. Para deficientes físicos, as novas tecnologias da informação e comunicação; se utilizadas adequadamente, podem trazer grandes contribuições, tanto no aspecto educacional, como no que diz respeito a condições de acesso.

As tecnologias de informação e de comunicação, estão cada vez mais interativas, além de permitir a interação dos seus usuários, possibilita escolhas e caminhos diferentes, e até mesmo, a produção e manipulação conjunta de conteúdos específicos. Nessa manipulação, é possível traduzir, codificando, decodificando, recodificando os conteúdos conforme a realidade e a necessidade do usuário, as suas histórias de vida e a cultura em que vivem. É evidente que as tecnologias de informação e de comunicação permitem uma comunicação mais dinâmica entre os participantes colaboradores da ação educativa. A aprendizagem colaborativa encara o usuário como elemento ativo no processo de aprendizagem e oferece a estes, grandes possibilidades de desenvolvimento de competências sociais e cognitivas pela troca de experiências e conhecimentos evidenciados.

2.2 Mobilidade e Acessibilidade

A mobilidade pode ser entendida como as condições do deslocamento humano e de bens pelas cidades, não importando a forma como o transporte é empregado: coletivo ou individual, motorizado ou não (Gomide & Galindo, 2013). Ela também envolve o planejamento urbano relacionado ao crescimento das cidades, e a preocupação em relação ao processo de circulação urbana das pessoas o qual, conseqüentemente, interfere na demanda de transportes, nas áreas de estacionamento e nos congestionamentos de veículos (Magagnin & Silva, 2008). Isso significa que a mobilidade urbana está profundamente relacionada à acessibilidade, ao direito e à necessidade que as pessoas têm de se locomover pelas cidades. Mobilidade é capacidade de

mover ou ser movido (Ikaputra e Sholihah, 2001). Mobilidade e acessibilidade são os principais componentes do sistema de transportes. Ilan Salomon e Patricia L. Mokhtarian (1998) definem mobilidade como uma exigência para todas as atividades que demandam de locomoção.

A definição de acessibilidade é constituída basicamente da ausência de limitações para o desenvolvimento das atividades humanas, abrangendo seus aspectos subjetivos (comportamento humano) como os aspectos objetivos (acessibilidade arquitetônica e urbana) exigindo que todas as suas dimensões atuem integralmente e de forma complementar (SASSAKI, 2009). Dentro do campo da construção civil, especificamente, a ABNT (2004) define “acessibilidade” como a “possibilidade e condição de alcance, percepção e entendimento para a utilização com segurança e autonomia de edificações, espaço, mobiliário, equipamento urbano e elementos” (NBR 9050/2004, p. 2).

Halden (2013) afirma que a acessibilidade urbana não é dependente de um fator apenas, mas da relação entre a realidade do espaço e a do indivíduo. Para ele, acessibilidade de uma pessoa é a capacidade do mesmo de poder alcançar diferentes possibilidades, enquanto a acessibilidade de um local é a facilidade deste em ser alcançado pelas pessoas, algo que pode ser feito de diferentes formas. O estudo da acessibilidade, para Halden (2013), se inicia a partir de três perguntas básicas – “quem, para onde, e como” – e depende da consideração dos seguintes aspectos: a) quem está sendo estudado – o indivíduo ou o lugar, pois a acessibilidade é um atributo de ambos; b) onde, ou a localização dos locais em que se as atividades de interesse para o indivíduo são desenvolvidas e, por fim, c) como, ou de que forma os fatores que dividem os indivíduos de locais de seu interesse e quais as possíveis soluções e conexões para superá-los ocorrem. Para que condução de diferentes pessoas a diferentes lugares, é preciso que as conexões entre ambos sejam bem resolvidas, em outras palavras, que os meios de transporte existentes sejam adequados a seu deslocamento.

No Brasil, a relação entre acessibilidade e mobilidade urbana é contemplada pela lei federal 12.587/12, a qual, institui as diretrizes da Política Nacional de Mobilidade Urbana. De acordo com esta lei, a acessibilidade corresponde a facilidade do deslocamento das pessoas dentro do espaço urbano; enquanto a mobilidade diz respeito à condição em que esse deslocamento, ocorre no espaço urbano. Dessa forma, a mobilidade é tratada como um aspecto físico que promove a autonomia que caracteriza a acessibilidade, contemplando desde o funcionamento de serviços de transporte público coletivo e sua integração com meios privados e não-motorizados, passando pela circulação viária, pelos polos geradores de viagens e áreas de estacionamento, até as áreas e horários de acesso e circulação restrita ou controlada.

Portanto, os elementos básicos da acessibilidade correspondem às pessoas, lugares e conexões, cuja sua aplicação exige tanto a definição e qualificação das necessidades do indivíduo, quanto dos elementos que o espaço deve oferecer para atendê-lo (HALDEN, 2013). A consideração às capacidades individuais de cada ser humano é a base para se elaborar soluções espaciais eficientes, uma vez que é por meio dela que se compreende a necessidade de cada um.

2.3 Aplicativos Colaborativos

Aplicativos colaborativos, também conhecidos como “*crowdsourcing*”, é um meio para resolução de problemas utilizando a contribuição de várias pessoas. Hoje em há vários exemplos populares de aplicativos e/ou serviços que se baseiam nesse modelo, entre eles, o

Foursquare, Wikipédia, YouTube, entre outros. Há duas particularidades que fazem os aplicativos colaborativos bastante atrativos: o baixo custo computacional e financeiro, no que diz respeito à obtenção de soluções e a solução de problemas que necessitam um alto grau de cognição, pois as soluções são produto do fator humano (PEDERSEN, 2013).

De uma forma geral, há três entidades envolvidas neste modelo:

- O dono do problema: aquele que sugere e expõe o problema a ser resolvido;
- A plataforma colaborativa: o responsável pelo gerenciamento e organização do problema e suas soluções geradas;
- O público: o que gera as soluções, incentivada por alguma recompensa (PEDERSEN, 2013).

O reconhecimento das entidades envolvidas neste modelo é mais subjetivo, pelo fato delas variarem de acordo com o ponto de vista com o qual um determinado serviço e/ou problema é analisado. Por exemplo, a organização do *YouTube* pode ser vista como o dono do problema, criar um acervo de vídeos, que recorre ao público para a criação do conteúdo.

Um sistema colaborativo é a constituição de um cyberspaço diferente. “Quem projeta e desenvolve sistemas colaborativos tem o poder de criar novas formas de trabalho e interação social, novos palcos para a convivência humana. Não basta conhecer bits e bytes, não é mais suficiente saber engenharia de software, é preciso também entender de gente, conhecer as características e necessidades de um novo ser humano digital e as novas formas de trabalho e organização social” (PIMENTEL e FUKS, 2011).

Jeff Howe defende que, na internet, a junção colaborativa de um grande número de pessoas não especializadas em um determinado assunto poderia ser altamente eficaz para a produção de ideias e projetos acerca deste mesmo assunto. A visão do autor tem um enfoque empreendedor, pois descreve o fenômeno segundo a perspectiva de modelo de negócio em que uma empresa propõe um desafio e abre o chamado para que o público ofereça soluções, de modo que as melhorias ideias sejam recompensadas e utilizadas (BITTENCOURT, 2014). A ideia de participação em rede por meio das novas tecnologias da informação, promove o engajamento do cidadão em uma ação de interesse público, exercendo sua cidadania, tornando a internet a serviço da sociedade (OLIVEIRA, 2014).

Pintado (2011) afirma que o envolvimento pode possuir uma motivação mais emocional, ou material. Howe (2008, p.12) afirma que “ao contrário do que diz a mentalidade convencional, o ser humano nem sempre se comporta seguindo padrões egoístas” os seres humanos podem contribuir por pouco ou nenhum retorno financeiro, podendo ser atraídas pelo desejo de ajudar uma comunidade, de fazer um bem maior, pelo prazer de praticar seu ofício, de se superar, pelo prazer em cultivar os próprios talentos e partilhar o que conhecem, de acordo com o autor, “no contexto de *crowdsourcing*, a colaboração é a própria recompensa”. O termo colaboração precisa ser contextualizado para definir a relação desejada entre os participantes [Brna, 1998]. Colaborar é o simples fato de pessoas trocarem informações e cooperarem entre si para produzir e manipular informações. O processo de colaboração é visto a partir dos seguintes pontos:

- Comunicação: é a base para a qualquer sistema colaborativo, pois todos os envolvidos precisam de uma forma para a troca de mensagens e informações. A troca de informações é um dos seus principais objetivos;

- Coordenação: é a execução de um processo qualquer, realizada por um grupo de pessoas que visam chegar a um produto final, que pode ser uma tabela, uma reportagem, um software ou qualquer tipo de artefato digital;
- Cooperação: os membros do grupo executam tarefas individualmente e depois combinam os resultados parciais para obter o resultado final.

Para Brabham (2013), aplicativos colaborativos (ou *crowdsourcing*) podem ser agrupados em quatro tipos de tipologia, baseadas nas soluções de problemas:

- Descoberta de conhecimento e gestão – direcionado para problemas na gestão da informação, na qual determinado grupo ou organização buscam unir várias pessoas para a obtenção de informações. É útil para gerar recursos coletivos.
- Inteligência humana distribuída em multitarefas – quando uma empresa ou organização possui um aglomerado de informações e se utiliza de um grupo de pessoas para realizar o processamento e/ou análise dessas informações.
- Pesquisa *broadcast* – voltado para o uso de um aglomerado de pessoas com o intuito de buscar uma solução para um determinado problema que tenha uma resposta completa e objetiva.
- Pontos-vetados de produção criativa – relacionado ao uso das multidões para a obtenção de uma solução de um problema que possuam uma resposta subjetiva ou que dependa de um apoio público.

As técnicas de sistemas colaborativos podem ser classificadas a partir de duas dimensões, a forma em que os dados são capturados e ao grau de participação do usuário. As informações podem ser obtidas a partir da extração de materiais enviados pelos usuários, os dados contidos nesse material podem ser usados sem que o usuário saiba explicitamente (Chatfield e Brajawidagda, 2014). O usuário pode também fornecer de forma consciente as informações de forma voluntária, tendo plena ciência do uso da mesma (Hirata et al., 2015).

Tendo em vista que quantidade de pessoas com acesso a internet no Brasil tem aumentado bastante nos últimos anos, 57% da população brasileira tem acesso a internet (Banco Mundial, 2014), nota-se que aplicativos colaborativos aberto ao público possui uma grande possibilidade de obter um grande de informações.

2.4 Aplicativos Móveis

Segundo dados da 28ª Pesquisa Anual de Administração e Uso de Tecnologia da Informação nas Empresas, realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP), o Brasil possui hoje cerca de 198 milhões de *smartphones* em uso, um crescimento de 17% comparados com os dados obtidos em 2016. O mesmo estudo ainda afirma que nos próximos dois anos, haverá um crescimento de 19% em relação aos dados atuais.

A demanda por aplicações móveis vêm crescendo junto com o aumento da quantidade de dispositivos móveis no mercado (CEVALLOS, 2014). A princípio, a única forma de havia para a criação de tais aplicações, era o desenvolvimento específico para cada plataforma, como por exemplo, o sistema iOS, que tem por padrão a linguagem *Objective-C*, ou a *Android*, que utiliza a linguagem java por padrão (HEITKÖTTER; HANSCHKE; MAJCHRZAK, 2013). Essa forma de desenvolvimento é conhecida como desenvolvimento nativo, o qual é projetado e concebido para uma plataforma específica (PREZOTTO; BONIATI, 2014). Nela, todas as

funções da plataforma são disponibilizadas sem restrições, elas também possuem seus próprios padrões de interface gráfica assim experiência do usuário específicas.

Com o grande aumento da demanda de aplicativos, veio a necessidade do rápido desenvolvimento dos mesmos, para todas as diferentes plataformas disponíveis no mercado, o que demanda muitos recursos, uma vez que era necessário manter equipes com conhecimentos diferentes para cada plataforma (PREZOTTO; BONIATI, 2014). A partir desse novo cenário, o desenvolvimento ágil de aplicações móveis que pudessem ser usadas em várias plataformas surgiu, o que ficou conhecido como desenvolvimento “multiplataforma”, que consiste em uma página web embutida dentro de um aplicativo nativo, o que permite a execução de um mesmo código que é executado e mostrado em um container dentro de um aplicativo nativo (STARK, 2010; HEITKÖTTER; HANSCHKE; MAJCHRZAK, 2013).

Com método de desenvolvimento de aplicativos móveis é possível o desenvolvimento de aplicações para diferentes plataformas sem a necessidade de possuir equipes especializadas para cada plataforma. No entanto, não há um padrão de interface ou de experiência de usuário que possa ser usado em várias plataformas ao mesmo tempo, o que pode causar uma sensação ruim na usabilidade e experiência de uso do aplicativo por parte do usuário (CORRAL; JANES; REMENCIUS, 2012).

2.4.1 Desenvolvimento de aplicativos móveis

El-Kassas et al. (2015) afirma que, o desenvolvimento de aplicativos móveis e o desenvolvimento dos demais tipos de software se diferenciam pelas particularidades e limitações que a primeira possui. Fatores como a capacidade do dispositivo móvel, mobilidade, especificações dos dispositivos, design e usabilidade.

2.4.1.1 Desenvolvimento Nativo

Os aplicativos nativos são desenvolvidos a partir do uso de linguagens e ferramentas específicas de cada plataforma, com o uso de frameworks a *SDK's* providos pelas mesmas. Esses aplicativos são incorporados apenas nos dispositivos do sistema alvo (EL-KASSAS et al., 2015). Se for necessário implementar o aplicativo para várias plataformas, essa aplicação deve ser reescrita para cada uma delas separadamente.

Tendo em vista que o aplicativo nativo é desenvolvido em seu próprio ambiente, ele deve seguir padrões visuais, técnicos e de usabilidade próprio de cada sistema.

2.4.1.2 Desenvolvimento Híbrido

Como dito anteriormente, para que aplicações desenvolvidas de forma nativa possam ser usadas em diferentes plataformas, deve ser feita uma aplicação para cada forma, o que demanda tempo, custos financeiros para as variadas tecnologias que envolvem as plataformas alvo. Para solucionar tal problema, temos em mão as soluções híbridas, que permite implementar aplicações que podem executar em diferentes plataformas usando o mesmo código (EL-KASSAS et al., 2015).

As aplicações híbridas surgiram a partir da combinação de tecnologias web, com o acesso às funcionalidades nativas (HEITKÖTTER; HANSCHKE; MAJCHRZAK, 2013). De

uma forma resumida, aplicações híbridas são aplicações web embutidas em um aplicativo nativo.

Nas próximas subseções serão descritas algumas das tecnologias utilizadas para o desenvolvimento multiplataforma.

2.4.1.3 Cordova

O Cordova é uma plataforma para desenvolvimento de aplicativos móveis, e ele concede às aplicações híbridas acesso às funcionalidades nativas dos dispositivos, como GPS, câmera, sensores. Contudo, o Cordova é capaz de apenas de dar funcionalidades nativas aos aplicativos híbridos, mas é incapaz de reproduzir a usabilidade e aparência dos dispositivos nativos (BEZERRA; SCHIMIGUEL, 2016).

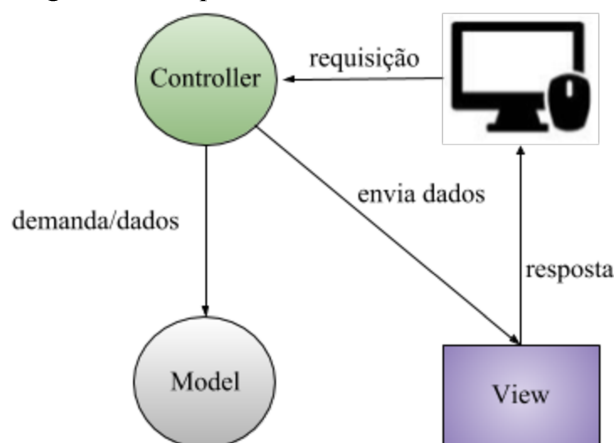
Visando suprir essa necessidade, diversos frameworks foram desenvolvidos em cima do Cordova para complementar, disponibilizando bibliotecas HTML e CSS para a criação de um *frontend* que possa oferecer uma experiência e usabilidade semelhantes de uma aplicação nativa (BEZERRA; SCHIMIGUEL, 2016).

Essas funcionalidades nativas podem ser acessadas através de *plugins*, que o Cordova possui. Elas são instaladas no projeto do aplicativo via CLI (BEZERRA; SCHIMIGUEL, 2016).

2.5 Padrão MVC

O MVC (*Model-View-Controller*) é um padrão de arquitetura de software que separa a aplicação em três camadas. A camada responsável por manipular os dados (*Model*), a camada que interage com o usuário (*View*) e a camada de controle (*Controller*). A figura abaixo, demonstra a relação entre as três camadas.

Figura 2 – Esquema básico do modelo MVC



Fonte: do autor, 2017

A *model* é a responsável pela manipulação de dados, suas leituras, escritas e validações. A *view* é camada de interação com o usuário, onde é feita a exibição das respostas. O *controller* é recebe todas as requisições vindas do usuário, os seus métodos (*actions*) são responsáveis por cada página, com a função de controlar qual *model* e *view* será exibido.

2.6 Frameworks

Um *framework* é um conjunto agregado de elementos de *software* (classes, objetos e componentes) que trabalham em unidos para oferecer uma arquitetura reutilizável para um grupo de aplicações relacionadas (Schmidt et al., 2004). Eles podem trazer uma grande quantidade de benefícios para de desenvolvimento de sistemas. Para Nash (2003), a maior vantagem é a velocidade no processo desenvolvimento. Quando a barreira curva de aprendizado é ultrapassada, o tempo gasto para o desenvolvimento de uma aplicação web pode se tornar consideravelmente bem menor se comparado ao desenvolvimento da mesma aplicação sem o uso de um framework.

Para Markiewicz e Lucena (2001), os *frameworks* devem ser encarados como uma saída no desenvolvimento de aplicações quando os requisitos do projeto sofrem mudanças constantemente e rapidamente, e além disso, um framework pode contribuir na padronização do processo de desenvolvimento, tornando o processo mais a produtivo e a manutenção do código mais fácil (Wong et al.,2006).

2.6.1 AngularJS

O HTML é a principal linguagem de marcação para a criação de páginas estáticas na *web*, no entanto, não há suporte para a exibição de conteúdos estáticos. O *AngularJS* é um framework que se utiliza da linguagem *JavaScript*, e que segue o padrão MVC *open source*, mantido pelo Google, o qual possibilita complementar a sintaxe do HTML para a o desenvolvimento de páginas dinâmicas (BEZERRA; SCHIMIGUEL, 2016). A partir dele, a utilização de *tags* não nativas do HTML se tornam possíveis, facilitando o gerenciamento de conteúdos ainda não dispostos na tela (GOOGLE, 2016).

2.6.2 Ionic Framework

O *Ionic* é um *framework open source* com a funcionalidade de auxiliar no desenvolvimento de aplicativos híbridos utilizando tecnologias web como HTML, CSS e *JavaScript*.

Ele foi criado em cima do Cordova e *AngularJS*, e a partir dele é possível desenvolver aplicativos móveis para diferentes plataformas (*Android*, iOS, entre outros) com apenas um código. O *Ionic* ainda conta com ferramentas para o auxílio do desenvolvimento dos aplicativos (DRIFTY, 2016c).

2.6.2.1 Arquetetura do projeto Ionic

O projeto *Ionic* é dividido em cinco camadas, que serão listadas e descritas a seguir:

- *Views*: é a camada de apresentação, a parte visível para o usuário. Pode ser comparada ao *view* do padrão MVC. São arquivos HTML que são chamados e inseridos quando necessário (DRIFTY, 2016d);
- *Controllers*: também pode ser comparada com a camada *controller* do padrão MVC. Ela é responsável pelo controle do fluxo de dados e lógica da aplicação.

Ela apresenta as *views* para o usuário, e chama a camada de dados para exibir as informações na tela para o usuário (DRIFTY, 2016d);

- *Data (Services/Factories)*: correspondente a camada *model* do padrão MVC. É quem encapsula os dados da aplicação, ela responde às requisições da *controller* com os dados usados para a criação da *view* que serão exibidos para o usuário (DRIFTY, 2016d);
- *App Configuration*: esta camada, liga as controladoras às suas interfaces por meio de rotas. Também é possível criar rotas padrões, para o caso de não haver nenhuma rota que esteja sendo identificada, o que poderia fazer o sistema quebrar (DRIFTY, 2016d);
- *Directives*: essa camada especifica comportamentos em elementos de uma página HTML, ou seja, elas são um elemento ou um atributo que podem iniciar um comportamento específico definido pelo desenvolvedor (DRIFTY, 2016d).

2.6.3 Laravel Framework

Laravel Framework é um *framework* utilizado para o desenvolvimento *web*, que utiliza a arquitetura MVC e tem como principal característica ajudar a desenvolver aplicações seguras e performáticas de forma rápida, com código limpo e simples, já que ele incentiva o uso de boas práticas de programação e utiliza o padrão PSR-2 como guia para estilo de escrita do código.

Para a criação de interface gráfica, o Laravel utiliza uma *engine* de *template* chamada *Blade*, que traz uma gama de ferramentas que ajudam a criar interfaces bonitas e funcionais de forma rápida e evitar a duplicação de código.

Para se comunicar com um Banco de Dados o Laravel utiliza uma implementação simples do *ActiveRecord* chamada de *Eloquent ORM*, que é uma ferramenta que traz várias funcionalidades para facilitar a inserção, atualização, busca e exclusão de registros. Com configuração simples e pequena e com pouco código podemos configurar a conexão com Banco de Dados e trabalhar com ele.

2.7 Linguagem de Programação PHP

PHP, *HyperText Preprocessor*, é uma linguagem de *open source*, desenvolvida especialmente para *web*. Nascida em 1994 como um projeto de Ramus Lerdorf, com o objetivo de monitorar os acessos à sua página *web* (PHP.NET, 2016). Essa linguagem baseia-se em C, Java e Perl, e permite a criação de sites *web* dinâmicos com acesso a banco de dados (HACKENHAAR, 2010).

O PHP pode ser escrito junto a linguagem de marcação HTML, em outras palavras, ele embute o PHP em meio ao código HTML, para que os processos possam ser enviados ao servidor, tendo a vantagem de não expor o código para o cliente.

Atualmente a linguagem PHP se encontra atualmente em 9º lugar, como uma das linguagens mais utilizadas no mundo (TIOBE, 2016).

Sendo assim, levando em consideração o conhecimento adquirido na universidade sobre a linguagem e observando a sua colocação como uma das linguagens mais utilizadas para a programação, esta será a linguagem a ser utilizada para o desenvolvimento do webservice da aplicação proposta neste trabalho.

2.8 Web Service

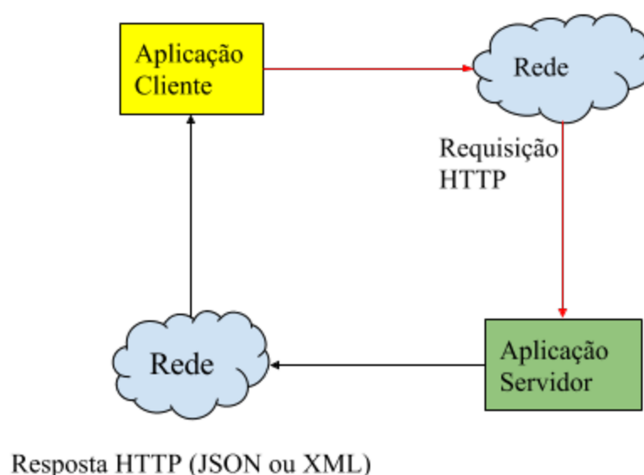
O Web Service é um sistema desenvolvido para operar entre duas máquinas dentro de uma rede, comunicando-se de forma transparente. Essa tecnologia tornou possível a interação entre sistemas diferentes e em que podem ser executados em plataformas diferentes.

Uma das suas principais características, é a possibilidade do uso de diferentes formas de transmissão de dados pela rede. A arquitetura *Web Service* pode atuar com diversos protocolos, como HTTP, FTP, REST entre outros [W3C 2004].

2.8.1 REST

REST, Representational State Transfer, é uma abstração da arquitetura da *World Wide Web*, um estilo arquitetural que basicamente é um conjunto coordenado de restrições arquiteturais aplicadas a componentes, conectores e elementos de dados dentro de um sistema de hipermídia distribuído.

Figura 3 - Troca de informação entre cliente e servidor com a tecnologia *REST*



Fonte: do autor, 2017

Na figura 3 acima, a Aplicação Cliente realiza uma requisição HTTP com as informações necessárias para executar uma operação específica para a Aplicação Servidor. A Aplicação Servidor realiza o processamento da requisição e retorna para a Aplicação Cliente uma resposta HTTP contendo um arquivo XML ou uma *String* JSON, podendo conter desde uma mensagem simples a um conjunto complexo de informações.

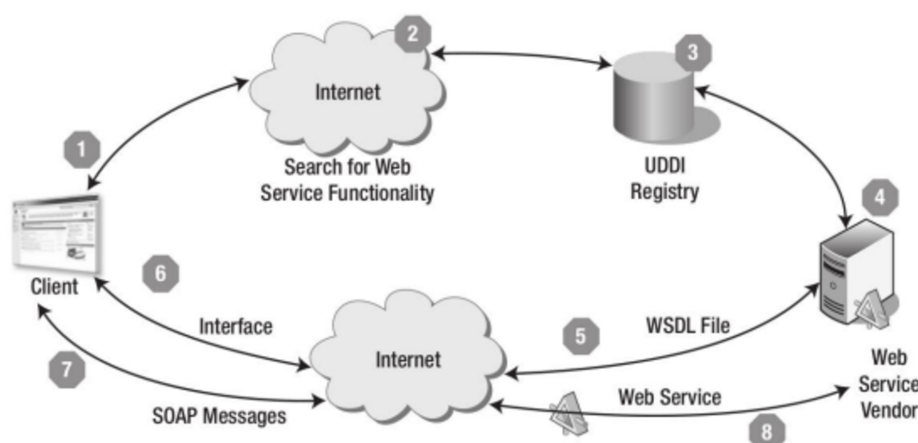
Um ponto importante dentro da arquitetura REST, é que não há um descritor do funcionamento do serviço. A Aplicação Cliente já conhece o que a Aplicação Servidor precisa receber, o que torna o processo de implementação mais simples.

O uso do REST é recomendado quando se há pouca limitação de recursos e de largura de banda. Também quando se tem um formato qualquer de retorno, de tal forma que qualquer navegador possa ser utilizado, pelo fato do REST adotar os padrões de chamada do HTTP (GET, PUT, POST, DELETE), que são suportados pela grande maioria dos navegadores.

2.8.2 SOAP

SOAP (*Simple Object Access Protocol*), é uma tecnologia que utiliza XML especificada pelo *World Wide Web Consortium* para o desenvolvimento de *web services*. É um protocolo leve que tem como objetivo a troca de informações estruturadas em um ambiente descentralizado (W3C, 2015). Como ele foi projetado para ser um protocolo de comunicação da camada de aplicação, é possível utilizar os protocolos HTTP ou SMTP para o transporte de suas mensagens através da Internet. Qualquer mensagem SOAP deve ser encapsulada por um envelope, que deve conter um cabeçalho (*header*) e um corpo (*body*), sendo que este último possui de fato a informação a ser transmitida (SILVA, 2016).

Figura 4 – Ciclo de vida de um serviço SOAP



Fonte: Peiris (2007)

Dentro de uma *web service* SOAP, há utilização de protocolos complementares, WSDL (*Web Services Description Language*) e o UDDI (*Universal Description, Discovery and Integration*), os dois são baseados em XML (SILVA, 2016). O primeiro protocolo possui como característica, a especificação da interface do serviço, métodos, tipo de dados, mensagens trocadas, entre outros, enquanto o segundo funciona como um catálogo de serviços e fornece um meio para a descoberta de nova *web services*. Na Figura 4 é possível observar como ocorrem as interações entres estes protocolos citados, o cliente busca um serviço acessando a um UDDI, em seguida baixa o WSDL do provedor escolhido e por último acessa à *web service* enviando mensagens SOAP (SILVA, 2016).

2.9 Google Maps

Google Maps é o serviço de mapeamento da Google, um serviço de pesquisa e visualização de mapas e imagens de satélite, gratuito na web, capaz de fornecer informações em tempo real. Além de poder ser usado como GPS (GOOGLE MAPS, 2017).

Ele possui uma API pública que permite qualquer desenvolvedor incorporar esse serviço a sua aplicação, permitindo a sua customização e personalização. A API possui disponibilidade para as seguintes plataformas: *Web*, *Android* e *iOS*. Essas APIs de plataformas nativas são complementadas pelo conjunto de serviços web HTTP da Google (GOOGLE MAPS, 2017).

Dentre as várias funções da *API*, a que mais será utilizada nesse projeto é a de geocodificação. Ela é o processo de conversão de um endereço em uma coordenada geográfica, que pode ser usada para inserir marcadores em um mapa ou posicionar o mesmo. O serviço inverso é também possível. Esses serviços são acessados a partir de requisições HTTP (GOOGLE MAPS, 2017).

3. TRABALHOS RELACIONADOS

Este capítulo abordará alguns outros aplicativos encontrados durante as pesquisas para esse projeto, que tinham como público-alvo, pessoas com deficiência. O objetivo é compreender e analisar as mais variadas abordagens na área de desenvolvimento de sistemas voltados para a acessibilidade e mobilidade. Alguns critérios foram escolhidos para selecionar os trabalhos relacionados, que serão listados a seguir:

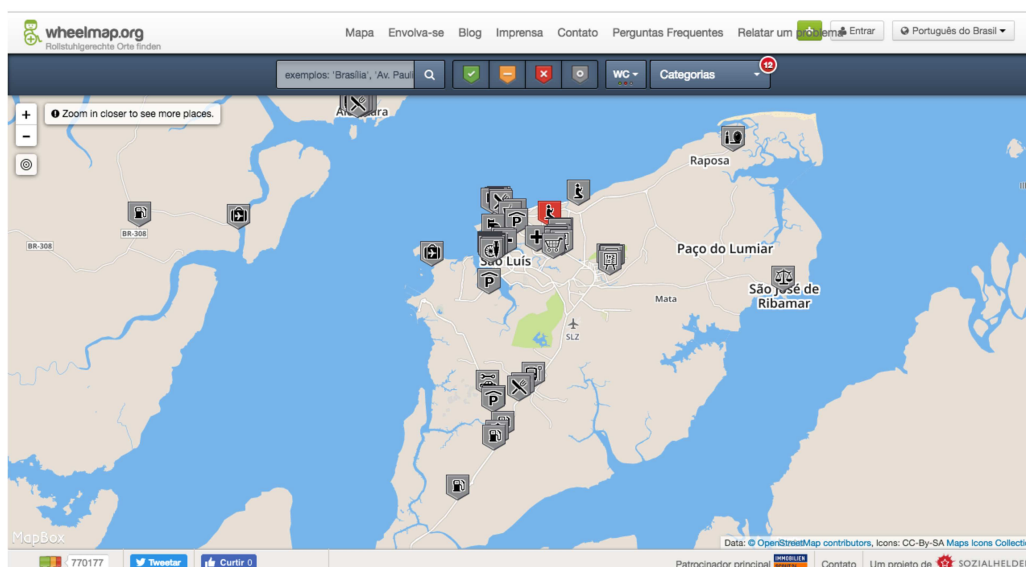
- Uma aplicação móvel como produto final;
- Possuir pessoas com deficiência locomotora como público-alvo;
- Um aplicativo colaborativo.

3.1 Wheelmap

Wheelmap é um aplicativo (móvel e *web*) criado pela organização não governamental alemã *Sozialhelden e.V* para encontrar locais acessíveis para cadeirantes. O mapa possui uma funcionalidade similar ao *Wikipedia*, qualquer pessoa pode contribuir e marcar lugares públicos ao redor do mundo de acordo com a acessibilidade (*Wheelmap*, 2017). O critério de marcação se baseia no semáforo:

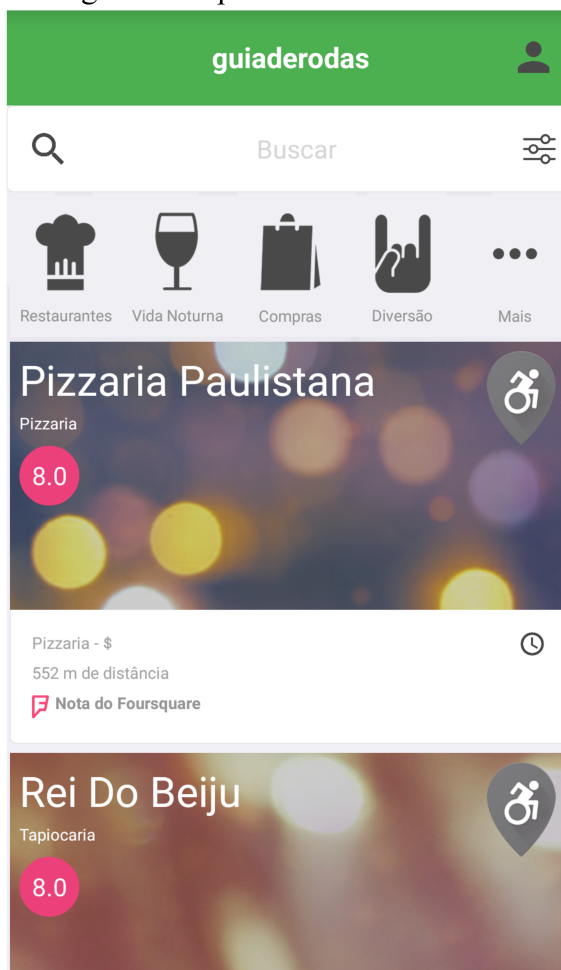
- Verde: total acessibilidade;
- Amarelo: média acessibilidade;
- Vermelho: nenhuma acessibilidade;
- Cinza: quando o local não possui nenhuma avaliação.

Figura 5 – Versão web do *Wheelmap*



Fonte: <https://wheelmap.org>, 2017

Figura 7 – Aplicativo Guiaderodas

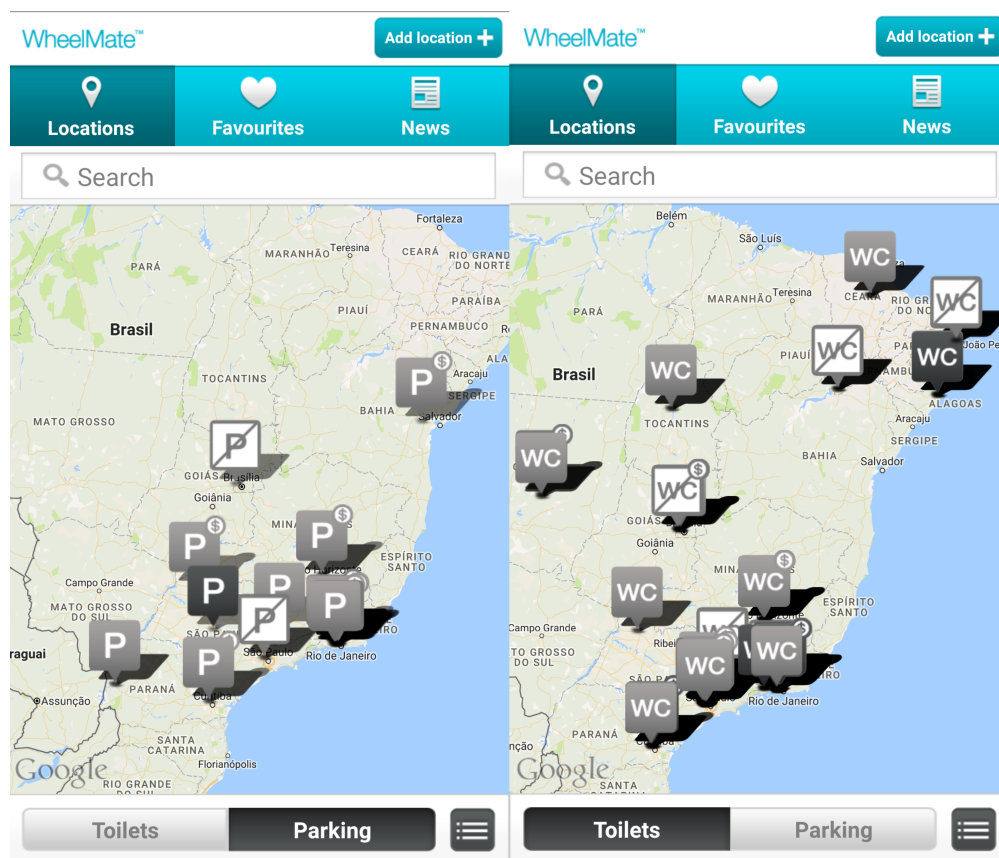


Fonte: Guiaderodas, 2017

O aplicativo é baseado no Foursquare, o que o faz ter uma abrangência mundial. Estabelecimentos do mundo inteiro podem ser avaliados, desde que o local esteja inserido na base do *Foursquare* (Guiaderodas, 2017). A avaliação dos locais se dá de forma simples, um pequeno questionário é feito, junto a um comentário no final.

3.3 WheelMate

WheelMate é um aplicativo desenvolvido pela empresa dinamarquesa *Coloplast*, e que tem como objetivo mapear banheiros e estacionamento que possuam acessibilidade. Nela o usuário pode inserir um novo local, informando se ele é um banheiro ou estacionamento, e em seguida inserir informações mais detalhadas, e também é possível realizar um comentário.

Figura 8 – Telas do aplicativo *WheelMate*

Fonte: WheelMate, 2017

O *WheelMate* possui versões para dispositivos *Android* e *iOS*, além de uma versão *web* do aplicativo, com as mesmas funcionalidades. Todas as localidades são adicionadas e verificadas pelos usuários, o que o caracteriza como um aplicativo colaborativo.

3.4 Comparação dos aplicativos

A seleção dos trabalhos relacionados considerou as necessidades encontradas durante esta pesquisa. A tabela 1 apresenta um comparativo entre as características dos trabalhos selecionados e o "Rota de Rodas" que será apresentado detalhadamente nos próximos capítulos.

O "*Wheelmap*" possui uma proposta semelhante ao Rota de Rodas, em relação a distribuir os locais no mapa e a partir de aí ver as avaliações ou criar uma nova avaliação do mesmo local. O grande diferencial desse aplicativo é a existência de uma versão *web* da mesma. Nele também é possível inserir novos locais, mas para isso é necessário registrar-se em um outro sistema, para ter acesso a tal funcionalidade. Por fim, o *Wheelmap* se restringe apenas a locais fechados, sendo impossível a inserção de locais abertos como ruas e avenidas. Nele, a avaliação ocorre apenas indicando se o local é "acessível para cadeirante", "parcialmente acessível para cadeira de rodas" ou "não acessível para cadeira de rodas".

Tabela 1 – Comparação entre os aplicativos

Aplicativo	Sistema Operacional Android	Voltados para pessoas com deficiência física	Aplicativo colaborativo	Avaliação de Locais Fechados Segundo Acessibilidade	Avaliação de Locais Abertos Segundo Acessibilidade	Possibilidade de Inserir Novos Locais
<i>Wheelmap</i>	x	x	x	x	-	x
Guiaderodas	x	x	x	x	-	-
<i>WheelMate</i>	x	x	x	x	-	x
Rota de Rodas	x	x	x	x	x	x

Fonte: do autor, 2017

O “Guiaderodas” é um aplicativo para avaliação de diversos locais. Nele o usuário ver, ao iniciar o aplicativo, uma lista com os locais mais próximos a ele, baseado na sua posição obtida pelo GPS do dispositivo. A partir daí o usuário pode avaliar os locais com um pequeno e simples formulário. No entanto, a funcionalidade de inserir novos locais é inexistente no guiaderodas, uma vez que todas as informações vêm de um banco de dados uma outra aplicação.

O *WheelMate* restringe-se apenas a avaliação de locais específicos, estacionamentos e banheiros. Ele exhibe todas as informações em um mapa, e os usuários podem selecioná-los e então realizar sua avaliação e deixar um comentário. Há a opção de inserir novos locais, mas não é possível inserir fotos do local juntamente com a avaliação.

4. ARQUITETURA DO SISTEMA

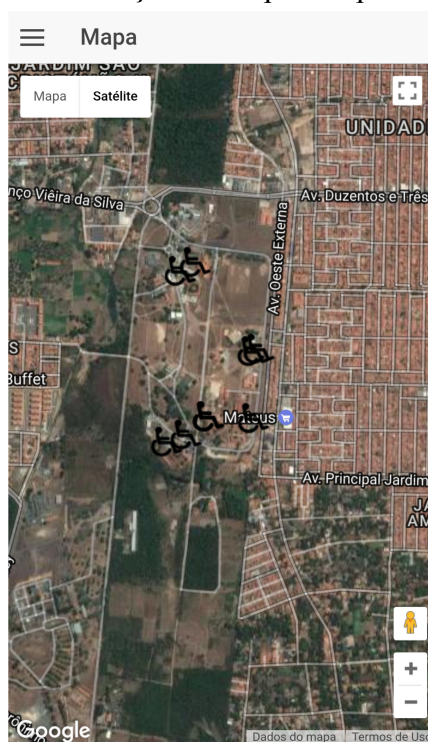
Pressman (2006), define a arquitetura de um sistema como a representação de uma estrutura para a composição de um sistema em termos de seus componentes computacionais, suas propriedades e interações. A arquitetura do software (ou sistema) tem a sua importância pelo fato de ela ser um facilitador da comunicação entre todas as partes envolvidas no desenvolvimento de um software, e também por focar nas definições iniciais de projeto, definições essas que são de grande importância em todo o processo de engenharia de software que se segue e, no sucesso final do sistema.

Nesta sessão serão utilizadas técnicas de modelagem para definir a arquitetura do projeto "Rota de Rodas", os diagramas seguirão técnicas baseadas a UML (*Unified Modeling Language*).

4.1 Rota de Rodas

Rota de Rodas é um aplicativo colaborativo, onde os mesmos poderão avaliar locais segundo seu nível de acessibilidade para deficientes locomotores. Os locais são divididos em duas categorias: locais internos (prédios e centros da universidade) e locais externos (ruas e vias de acesso da universidade). Para cada categoria há um questionário que possui 5 opções como resposta: péssimo, ruim, regular, bom e ótimo.

Figura 9 – Exibição do mapa no aplicativo



Fonte: do autor, 2017

Após concluir o questionário, é solicitado que se faça uma foto do local avaliado. Com a conclusão da avaliação de um local, o mesmo é exibido em mapa dentro do aplicativo, baseado

nas suas coordenadas geográficas. É possível também ver todas as avaliações feitas em um local, assim como as fotos feitas do mesmo e acrescentar uma nova avaliação.

4.2 Definição do Sistema

4.2.1 Plataforma

O "Roda de Rodas" será desenvolvido para sistemas *Android* voltados para *smartphones* e *tablets*. As justificativas para isso são os números que o mercado móvel apresentou nesses últimos anos, onde houve uma grande hegemonia dos dispositivos *Android*, com mais de metade da quota de mercado na venda de dispositivos móveis. Segundo a Kantar (2017), o número de dispositivos *androids* nas mãos dos usuários correspondem a 92,1% em março de 2017 no Brasil. E esse número só tende a crescer.

4.2.2 Framework Usados

4.2.2.1 Aplicativo

Para desenvolvimento do protótipo do aplicativo foi escolhido o uso do *Ionic Framework*. Ele foi escolhido pelo fato de a curva de aprendizagem ser pequena, e pela facilidade de se trabalhar para várias plataformas. Além do fato de usar tecnologias web (*JavaScript*, HTML e CSS) como seu principal alicerce, o que significa que ele trabalha com a parte gráfica. Para o acesso das funções nativas dos dispositivos, escolheu-se o uso do Apache Cordova.

O *Ionic Framework* também conta com um conjunto próprio de ferramentas que auxiliam no desenvolvimento de aplicativos móveis, como o *Ionic Creator*, que é uma ferramenta online para criação de protótipos funcionais, e onde é possível exportar o projeto para inserir as regras de negócio.

4.2.2.2 Webservice

Para o desenvolvimento do webservice que alimenta o aplicativo Rota de Rodas, o escolhido foi o Laravel Framework, um framework que segue o padrão MVC. A curva de aprendizado foi um fator determinante para a escolha dessa ferramenta, além de todas as suas vantagens que ela trás. Como o *Eloquent ORM (Object Relational Mapper)*, como ele é possível trabalhar com banco e dados e com suas principais funcionalidades (*insert, update, delete, select*, relacionamentos) de forma simples e elegante., códigos *SQL* não são usados. O uso do padrão MVC por parte do *framework*, é também um fator decisivo para a escolha do mesmo, uma vez que esse tipo de padrão torna o desenvolvimento de sistemas mais simples e ágeis.

Tabela 2 – Funções *Eloquent ORM*

Descrição	Função
Listagem de Locais	Local::all()
Buscar local específico	Local::find(\$id)
Excluir local	Local::delete(\$id)

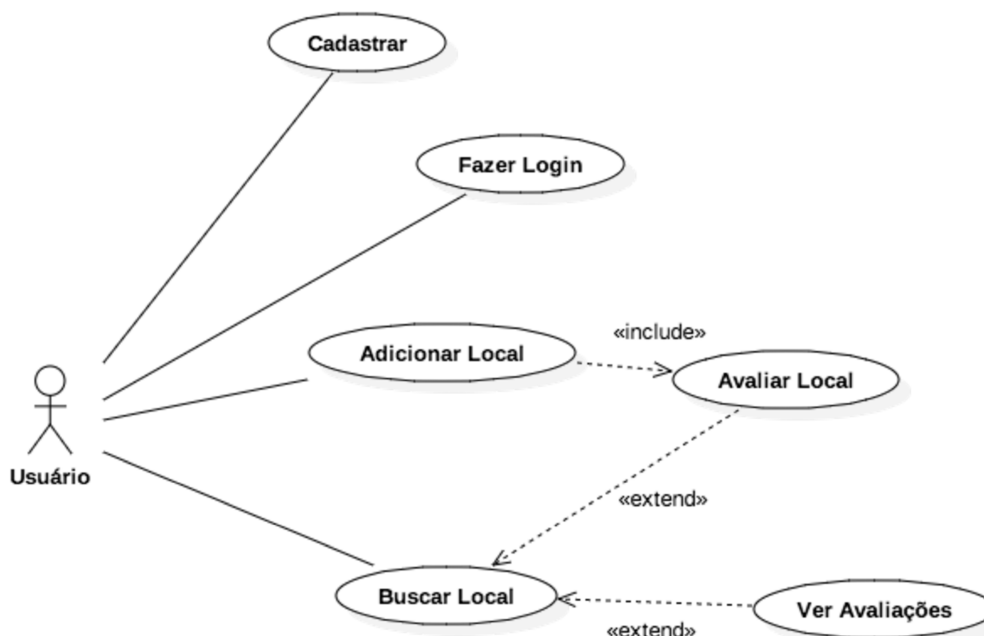
E dentre todas as vantagens oferecidas pelo laravel, temos a de que é fácil de encontrar soluções na internet, este framework possui uma grande, e ativa comunidade, além de uma documentação completa em seu *website*.

4.3 Modelagem do Sistema

4.3.1 Caso de Uso

A modelagem de caso de uso tem como objetivo identificar os recursos e funcionalidades que o usuário espera que o sistema suporte, no entanto, ele não deve revelar os detalhes a respeito da implementação dos recursos (Pender, 2004). Para Bezerra (2014), ele é uma representação das funcionalidades que podem ser observadas e dos elementos externos que interagem com ele através da representação com atores. Na figura abaixo é mostrado o diagrama de caso de uso do sistema rota de rodas.

Figura 10 – Diagrama de Caso de Uso do Sistema Rota de Rodas



Fonte: do autor, 2017

Os principais elementos do caso de uso são:

- Os atores, representados por bonecos feitos de traços;

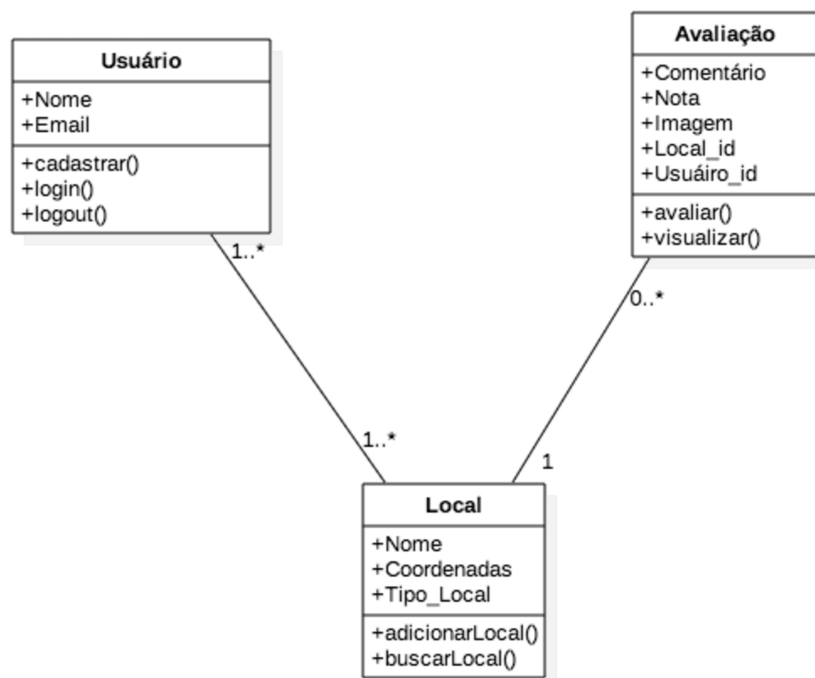
- As ações, representadas por caixas com bordas arredondadas, e que tem por função representar o objetivo de uma ação;
- As relações, representadas por uma seta, e que representa a relação entre um ator e uma ação, ou entre duas ações.

A modelagem de caso de uso é parte importante no momento da especificação de requisitos, ele é responsável pela moldagem dos requisitos funcionais do sistema. Os casos de uso refletem a realidade do sistema, temos o ator usuário que interage com o sistema, ele realiza o próprio cadastro, assim como o *login*. Com registro no sistema feito, e após ter o *login* realizado, o usuário pode adicionar um novo local, o que trás consigo a função de avaliar esse novo local inserido. E ao buscar um local, o usuário pode realizar uma nova avaliação para esse local, ou apenas visualizar as avaliações já existentes nesse local.

4.3.2 Diagrama de Classes

Para Bezerra (2014) o diagrama de classes refere-se à descrição dos variados objetos encontrados em um sistema, e os seus relacionamentos. A sua representação se dá por retângulos, e possuem um nome, atributos e métodos. Segundo Guedes (2011) o diagrama de classes é responsável por definir a estrutura das classes que serão usadas pelo sistema, nele é possível definir os atributos e métodos que cada classe possui, exibindo também como as classes se relacionam entre si.

Figura 11 – Diagrama de Classe do Sistema Rota de Rodas



Fonte: do autor, 2017

Para o sistema de do Rota de Rodas, foram definidas e modeladas três classes, como é possível ver na figura X acima. A classe usuário corresponde ao usuário do sistema, ele possui os métodos de cadastro (próprio), *login* e *logout* do aplicativo. A seguir temos a classe local,

que representa os locais já inseridos e que serão inseridos, ele possui os métodos de buscar um local existente e adicionar um novo. E por último a classe avaliação, responsável pela a avaliação dos locais, a mesma possui os métodos avaliar e visualizar.

5. PROTOTIPAGEM DO APLICATIVO ROTA DE RODAS

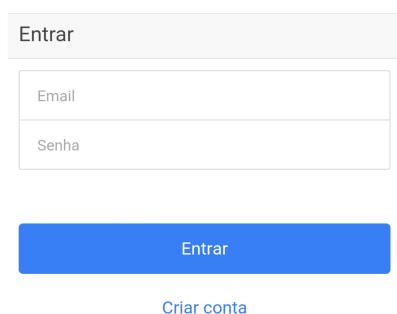
O uso de protótipos pode dar uma base para subsidiar tomadas de decisão e são uma forma de ganhar experiência prática, além de auxiliar a especificação de requisitos (Budde e Zullighoven, 1990). Os protótipos podem possuir alta ou baixa fidelidade, quando mais próximo do software real, maior é a sua fidelidade (Walker et al., 2002). Para Pressman e Maxim (2016), prototipação é a técnica de desenvolvimento de uma previa do sistema, essa técnica auxilia os desenvolvedores da aplicação e o usuário, a entenderem o que o sistema realmente precisa.

5.1 Elementos da tela do aplicativo

5.1.1 Tela de autenticação do aplicativo

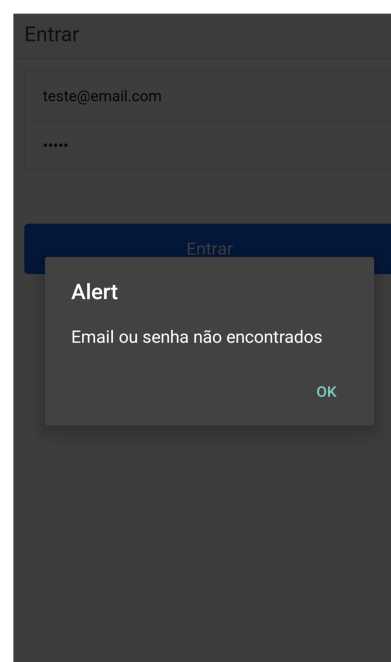
Nesta tela, o usuário poderá inserir suas informações de autenticação (e-mail e senha), para assim poder ter acesso às funcionalidades do aplicativo. Ao clicar no botão “Entrar” uma requisição do tipo *post* é enviada ao servidor via *http*, se as credenciais estão corretas o usuário é redirecionado para a tela do mapa do aplicativo, caso contrário, uma mensagem de erro é exibida, como mostrado na figura 13.

Figura 12 – Tela de autenticação



Fonte: do autor, 2017

Figura 13 – Tela de erro de autenticação



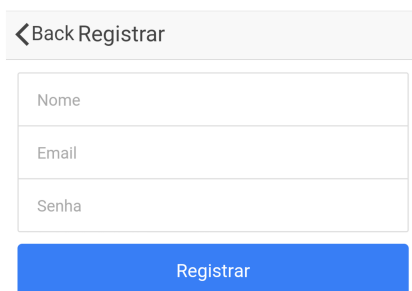
Fonte: do autor, 2017

Nesta mesma tela, caso o usuário ainda não tenha registro no aplicativo, ao clicar no botão “criar conta”, ele poderá ter acesso à tela de cadastramento, que será exibida no próximo tópico.

5.1.2 Tela de cadastro de novo usuário

Para usar o protótipo do aplicativo Rota de Rodas é necessário se registrar no sistema, para isso, é preciso inserir informações do usuário, como o nome, e-mail (que será usado para autenticação no aplicativo) e a senha.

Figura 14 – Tela de cadastro do aplicativo



A imagem mostra a tela de cadastro do aplicativo. No topo, há uma barra de navegação com um ícone de seta para trás e o texto "Back Registrar". Abaixo, há três campos de entrada de texto empilhados, rotulados "Nome", "Email" e "Senha". Na base da tela, há um botão azul com o texto "Registrar".

Fonte: Do autor

Após o registro concluído, o usuário é redirecionado para a tela de autenticação, onde o mesmo poderá inserir suas credenciais e se autenticar no aplicativo, e então utilizar as funcionalidades do mesmo, que serão descritas a seguir.

5.1.3 Menu do aplicativo

O menu é responsável pela navegação dentro do aplicativo, ajudando o usuário a encontrar as telas que ele quer e, por consequência, obter a informação que ele busca nesta tela. Na figura 8, é mostrado os menus disponíveis no aplicativo RRODAS.

Figura 15 – Menus do aplicativo

Menu	☰ Ma
Mapa	
Adicionar Local	
Perfil	

Fonte: Do autor

São exibidos 3 menus, dentro aplicativo: mapa, novo local e perfil. Ao clicar em cada um deles, o usuário é redirecionado para as respectivas telas, telas essas que serão abordadas detalhadamente nas próximas sessões.

5.1.3 Tela de novo local

A principal característica do aplicativo Rota de Rodas, é a possibilidade de adicionar os locais e avalia-los, cada local tem um nome a ser inserido, assim como um tipo e um comentário sobre o local. No aplicativo é possível avaliar qualquer lugar, seja uma praça, uma rua, avenidas, restaurantes, prédios. Ao adicionar um novo local, o usuário deve realizar uma avaliação do mesmo, respondendo perguntas objetivas que variam de acordo com o tipo de local selecionado. As opções de respostas são as mesmas para todas as perguntas: péssimo, ruim, regular, bom e ótimo.

Há depois tipos de locais no aplicativo, áreas internas (referentes a locais fechados, como prédios, restaurantes) e áreas externas (referentes a locais abertos, como ruas, avenidas, praças). Para cada tipo há perguntas específicas, baseadas na norma da ABNT NBR 9050, que trata da acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos. As áreas externas possuem as seguintes perguntas:

- Calçamento – qualidade da área destinada ao deslocamento;
- Autonomia de locomoção – sobre a necessidade de ajuda para locomoção no local;
- Rampas – sobre a disponibilidade, ou falta, de rampas no local.

Figura 16 – Formulário de novo local (área externa)

Novo Local	
Nome	
Tipo	Área Externa ▼
Calçamento	
Qualidade da calçada na qual o deslocamento é feito ▼	
Autonomia de locomoção	
É possível se locomover livremente, sem ajuda de alguém? ▼	
Rampas	
Qualidade das rampas disponíveis no local ▼	
Comentário	

Salvar Novo Local e tirar foto

Fonte: do autor, 2017

Para áreas internas há as seguintes perguntas:

- Calçamento – sobre a área de acesso ao redor do local;
- Entradas do prédio – a respeito das condições das entradas do local;
- Acesso à veículos – sobre a existência ou não de vagas especiais;
- Locomoção interna – sobre o espaço interno do local que é destinado à locomoção;
- Banheiros – sobre as condições de acessibilidade dos banheiros do local.

Figura 17 – Formulário de novo local (área interna)

Novo Local	
Nome	
Tipo	Área Interna (Prédios) ▼
Calçamento	▼
Área de acesso ao redor do prédio	
Entradas do Prédio	▼
Condições da(s) entrada(s) do local	
Acesso à veículos	▼
Vagas especiais para cadeirantes	
Locomoção interna	▼
Sobre o espaço interno do prédio, destinado à locomoção	
Banheiros	—

Fonte: do autor, 2017

Ao concluir a inserção de um novo local, o usuário é obrigado a tirar uma foto do mesmo, o qual será salvo em um servidor e ficará sempre disponível para visualização na área de avaliações, o qual será descrito a seguir.

5.1.4 Mapa

Uma das principais funcionalidades do aplicativo é a capacidade de visualizar em um mapa os locais inseridos, e partir de aí acessar todas as avaliações de um local e também adicionar uma nova avaliação nesse mesmo local. Cada local é identificado a partir de um marcador no mapa, representado por um desenho de uma pessoa em uma cadeira de rodas, como mostrado na figura 18.

Nesta tela é utilizada a *api* do *Google Maps* para a renderização do mapa. As coordenadas de todos os locais são buscadas na base de dados do servidor a partir de uma requisição *http* utilizando o verbo *get*. Através da *api* do *Google Maps*, são gerados os marcadores que são exibidos na tela, a partir das coordenadas obtidas de cada local. Ao clicar nos marcadores, janelas com a descrição do local, e links para as telas de “ver avaliações” e “nova avaliação” são exibidas, como mostrado na figura 19.

Figura 18 – Mapa



Fonte: do autor, 2017

Figura 19 – Mapa com detalhes do local



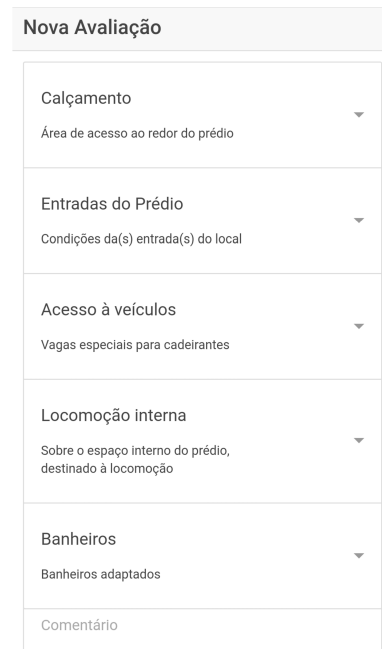
Fonte: do autor, 2017

Figura 20 – Avaliação de um local



Fonte: do autor, 2017

Figura 21 – Nova avaliação de um local



Fonte: do autor, 2017

Ao clicar em “Ver avaliações” (visto na figura 19) o usuário é redirecionado para a tela de avaliações, onde ele poderá visualizar a avaliação do local selecionado. Nesta é tela, é possível visualizar a avaliação geral do local, assim como as avaliações de cada pergunta do local, como é possível ver na figura 20. Já ao clicar em “Nova Avaliação”, o usuário poderá

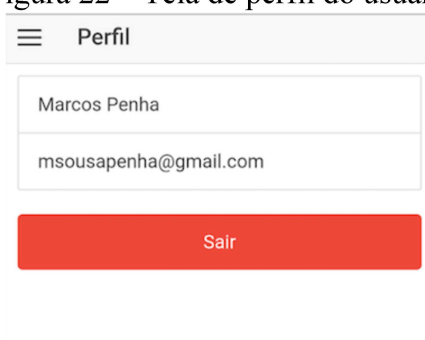
inserir uma nova avaliação para o local, respondendo as perguntas, realizando um comentário e por fim, enviando uma nova foto do local, essa tela é mostrada na figura 21.

A tela de nova avaliação segue os mesmos padrões da tela de novo local, com a exceção do nome do local, o qual não é necessário que seja informado pelo usuário. As perguntas são as mesmas utilizadas no formulário de novo local, baseadas na norma da ABNT NBR 9050.

5.1.5 Perfil de Usuário

Por fim, a tela de perfil do candidato, onde o usuário pode verificar suas informações pessoais cadastrados no aplicativo. Também é possível nessa tela, realizar o *logout* aplicativo, função essa que, após realizada com sucesso, redireciona o usuário para a tela de autenticação. Essa tela pode ser observada na figura 22 a seguir.

Figura 22 – Tela de perfil do usuário.



Fonte: do autor, 2017

6. CONCLUSÃO

Foi proposto neste trabalho o desenvolvimento de um aplicativo móvel colaborativo voltado para pessoas com deficiência física, onde os próprios usuários podem inserir e avaliar locais de acordo com o seu nível de acessibilidade, e ao mesmo tempo verificar quais locais ao seu redor possui melhor é acessível para pessoas com deficiência locomotora. Esta pesquisa foi motivada pelo fato de várias cidades no Brasil (incluindo São Luís – MA) possuírem uma baixa infraestrutura para pessoas com esse tipo de deficiência.

Para o alcance dos resultados esperados, primeiramente foi-se buscado um embasamento teórico sobre deficiência, acessibilidade e mobilidade, aplicativos colaborativos e a respeito de aplicativos móveis em geral, também foi buscado durante essa etapa ferramentas que fornecessem todos os meios necessários para o desenvolvimento do software proposto, onde foi constatado que o uso do *Ionic Framework* era a melhor opção para o desenvolvimento do aplicativo e o *Laravel Framework* para o desenvolvimento do *webservice*, pois ambos oferecem agilidade no desenvolvimento de softwares. E o *Ionic Framework*, especialmente, por se tratar de uma ferramenta para desenvolvimento de aplicações híbridas, ou seja, há a possibilidade de desenvolvimento para diferentes plataformas.

Durante as buscas por trabalhos relacionadas na área de aplicativos colaborativos para deficientes físicos, foram selecionados aqueles que correspondiam com os objetivos e características desse trabalho. E após, foi feita uma análise comparativa entre todos os aplicativos.

Um outro momento importante para o desenvolvimento de softwares é a modelagem e a definição do sistema, tais processos facilitam o desenvolvimento, dá mais robustez ao sistema e torna mais fácil a manutenção do software. Para o sistema “Rota de Rodas”, técnicas com base na linguagem UML foram usadas (diagrama de casos de uso e diagrama de classes). E após a etapa de modelagem, um protótipo foi criado para *smartphones Android*.

O aplicativo se propõe criar um ambiente interativo e colaborativo visando ajudar pessoas com deficiência física a buscar locais que estejam aptos a recebe-los, mostrando em um mapa gerado a partir da API do Google *Maps* todos os locais já avaliados, e dando a possibilidade de qualquer usuário inserir e avaliar um novo local, trazendo o que de melhor a internet e as novas tecnologias tem a oferecer: o acesso e o compartilhamento democrático do conhecimento e informação.

6.1 Trabalhos Futuros

O projeto seguirá em desenvolvimento, buscando melhorias e novas funcionalidades. Várias são as melhorias aplicáveis para o aplicativo. Neste primeiro momento temos as seguintes propostas:

- Uso *offline* do aplicativo;
- Traçar rotas a partir de locais avaliados;
- Adaptar para demais deficiências;
- Criar uma aplicação web.

REFERÊNCIAS

ABNT, NBR. 9050: 2004. **Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos [Internet]**, 2004.

BEZERRA, P. T.; SCHIMIGUEL, J. Desenvolvimento de aplicações mobile crossplatform utilizando phonegap. 2016. Disponível em: <<http://eumed.net/cursecon/ecolat/br/16/phonegap.html>>. Acesso: 17 mai. 2017.

Bittencourt, L. Colaboração em Massa (Crowdsourcing) na comunicação corporativa. Disponível em: <<http://www.bocc.ubi.pt/pag/bittencourt-filho-colaboracao-em-massa-crowdsourcing.pdf>>. Acesso: 08 jun. 2017.

BRABHAM, Daren C. Crowdsourcing, MIT Press. 2013.

BRASIL. Lei nº 7.853, de 24 de outubro de 1989. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/L7853.htm>. Acesso: 20 mai. 2017.

BRASIL. **LEI Nº 12.587**, de 3 de janeiro de 2012. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2012/lei/112587.htm>. Acesso: 18 mai. 2017.

BEZERRA, E. **Princípios de análise e projeto de sistemas com UML**. 3. ed. Rio de Janeiro: Elsevier, 2014.

BRNA, P.; BURTON, M. **The computer modelling the students collaborating in learning about energy**. Journal of computer assisted learning, 13:193-204, 1997.

BUDDE, R., ZULLIGHOVEN, H. **Prototyping revisited**”, **Proceedings of the 1990 IEEE International Conference on Computer Systems and Software Engineering**. TelAviv, Israel, p. 418-427, 1990.

CEVALLOS, E. A. **Case Study on Mobile Applications UX: Effect of the Usage of a Cross Platform Development Framework**. Tese (Master Thesis) — UNIVERSIDAD POLITÉCNICA DE MADRID, Madrid, jun. 2014. Disponível em: <http://oa.upm.es/30422/1/EMSE-2014-05_Esteban_Angulo-1.pdf>. Acesso em: 08 mai. 2017.

CHAPPELL, Tyler, J. **Java Web Services: Up and Running**, O’Reilly Media; 1 edition. 2002.

CHATFIELD, A.; BRAJAWIDAGDA, U. **Crowdsourcing hazardous weather reports from citizens via twittersphere under the short warning lead times of EF5 intensity tornado conditions**. In: HICSS, Waikoloa, USA. 2014.

CORRAL, L.; JANES, A.; REMENCIUS, T. Potential Advantages and Disadvantages of Multiplatform Development Frameworks - A Vision on Mobile Environments. Procedia

Computer Science, v. 10, p. 1202–1207, jan. 2012. ISSN 1877-0509. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S1877050912005303>>. Acesso: 08 mai. 2017.

DINIZ, Debora. **O que é deficiência**. 1º Edição. São Paulo: Brasiliense, 2007.

DRIFTY. About Ionic - Ionic Components. 2016. Disponível em: <<http://ionicframework.com/docs/v1/overview/#about>>. Acesso: 18 mai. 2017.

DRIFTY. Ionic Concepts - App Structure - Ionic Framework. 2016. Disponível em: <<http://ionicframework.com/docs/v1/concepts/structure.html>>. Acesso: 21 mai. 2017.

DOUGLAS, C. Schmidt; ANIRUDDHA, Gokhale; BALACHANDRAN Natarajan. **Frameworks: Why they are important and how to apply them effectively**. ACM Queue magazine, 2004.

EL-KASSAS, W. S. et al. Taxonomy of Cross-Platform Mobile Applications Development Approaches. Ain Shams Engineering Journal, 2015. ISSN 2090-4479. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S2090447915001276>>. Acessado em: 15/05/2017.

ESTELLES-AROLAS, E.; GONZALEZ-LADRON-DE-GUEVARA, F. **Towards an integrated crowdsourcing definition**. Journal of Information, 2012.

FIELDING, Roy Thomas. *Architectural Styles and the Design of Network-based Software Architectures*. Doctoral dissertation, University of California, Irvine, 2000.

GAMMA, et al. **Padrões de projeto: soluções reutilizáveis de software orientado a objetos**. Porto Alegre: Bookman, 2000.

GERHARDT, T.E.; SILVEIRA, D. T. **Métodos de pesquisa**. Universidade Aberta do Brasil – UAB/UFRGS. Curso de Graduação Tecnológica – Planejamento e Gestão para o Desenvolvimento Rural da SEAD/UFRGS. 1ª Edição. Editora da UFRGS, Porto Alegre, 2009. Disponível em: <<http://www.ufrgs.br/cursopgdr/downloadsSerie/derad005.pdf>>. Acesso: 13 jul. 2017.

GOMIDE, A. Á., GALINDO, E. P. **A mobilidade urbana: uma agenda inconclusa ou o retorno daquilo que não foi**. Estudos Avançados, 27(79), 27-39, 2013. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S010340142013000300003&lng=pt&tlng=pt>. Acesso: 08 mai. 2017.

GONÇALVES, R. F., GAVA, V. L., PESSÔA, M. S. P., SPINOLA, M. M. **Uma proposta de processo de produção de aplicações web**. Revista Produção, vol. 15, n. 3, p. 376-389, Set./Dez. 2005.

GOOGLE. AngularJS - Superheroic JavaScript MVW Framework. 2016. Disponível em: <<https://angularjs.org/>>. Acesso: 17 mai. 2017.

GUEDES, Gilleanes T. A. UML 2: **Uma abordagem prática**. São Paulo: Novatec, 2011.

Guia de rodas. Disponível em: <<http://www.guiaderodas.com/>>. Acesso: 17 mai. 2017.

HACKENHAAR, J.; ZANELLA, R.; CARDOSO, T. **Um comparativo entre php e jsp: definindo a melhor aplicação para o desenvolvimento de projetos web**. Revista iTEC, v. 1, p. 32 – 36, 2010

HALDEN, D. **The use and abuse of accessibility measures in UK passenger transport planning**. Disponível em: <<http://dhc1.co.uk/content/useandabuseonline.pdf>>. Acessado em: 08.06.2017

HEITKÖTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. **Comparing Crossplatform Development Approaches for Mobile Applications**. In: Web information systems and technologies. Springer, 2013. p. 120–138. Disponível em: <http://link.springer.com/chapter/10.1007/978-3-642-36608-6_8>. Acessado em 16/05/2017.

HERSH, M. A.; JHONSON, M. A. *Assistive Technology for Visually Impaired and Blind People*. [S.l.]: Springer, 2008. Acesso em: 22.11.2015.

HIRATA et al. **Flooding and inundation collaborative mapping - use of the crowdmap/ushahidi platform in the city of Sao Paulo – Brazil**. Journal of Flood Risk Management. 2015.

IKAPUTRA, Arif B. SHOLIHAN. **MOBILITY FOR ALL Towards Barrier Free Environment in Yogyakarta - Indonesia**. IATSS Research. 2001.

LARAVEL. **Laravel Documentation**. Disponível em: <<https://laravel.com>>. Acesso: 01 mai. 2017.

MARKIEWICZ, M. E.; LUCENA, C. J. **Object oriented framework development**. ACM Crossroads Student Magazine, [S.l.], 2001.

MAGAGNIN, R. C., SILVA, A. N. R. **A percepção do especialista sobre o tema mobilidade urbana**. Transportes, 16(1), 25-35, 2008. Disponível em:<<http://dx.doi.org/10.14295/transportes.v16i1.13>>. Acesso: 08 mai. 2017.

MEIRELLES, Fernando, S. **28ª Pesquisa Anual do Uso de TI**. Fundação Getúlio Vargas. 2017.

NASH, M. **Java Frameworks and Components: Accelerate Your Web Application Development**. Cambridge Universty Press, 2003.

OLIVEIRA, V, **O crowdsourcing a frente da mídia colaborativa e democrática: uma perspectiva cidadã para Web 2.0.** Disponível em: <<http://www.unicentro.br/redemc/2012/artigos/34.pdf>>. Acesso: 08 jun. 2017.

PEDERSEN, J. et al. **Conceptual foundations of crowdsourcing: a review of IS research.** Proc. 46th Hawaii Int. Conf. on System Sciences, (2013).

PENDER, Tom. **Uml, a Bíblia.** Tradução Daniel Vieira – Rio de Janeiro: Editora Campus, 2004.

PINTADO, Diego H. **Crowdfunding e a cultura de participação.** Universidade Federal do Rio Grande do Sul. 2011.

PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional.** 8. ed. Porto Alegre: AMGH, 2016.

PREZOTTO, E.; BONIATI, B. **Estudo de Frameworks Multiplataforma para Desenvolvimento de Aplicações Mobile Híbridas.** 2014

SALOMON I.; MOKHTARIAN, P. L. **What happens when mobility-inclined Market segments face accessibility-enhancing policies?** Transportation Research D3(3). 1998.

SASSAKI, R. K. **Inclusão: acessibilidade no lazer, trabalho e educação.** In: Revista Nacional de Reabilitação (Reação). São Paulo, Ano XII, mar./abr. 2009.

STARK, J. Building iPhone Apps with HTML, CSS, and JavaScript. O'Reilly Media, 2010. ISBN 978-0-596-80578-4. Disponível em: <<http://shop.oreilly.com/product/9780596805791.do>>. Acesso em 08 mai. 2017.

SILVA, Adilson Florentino da; CASTRO, Ana de Lourdes Barbosa de; BRANCO, M. C. M. C. A inclusão escolar de alunos com necessidades educacionais especiais: deficiência física. **Brasília: Ministério da Educação, Secretaria de Educação Especial**, v. 8, 2006.

TIOBE, the software quality company, 2016. Disponível em: <http://www.tiobe.com/tiobe_index>. Acesso em 26 jul. 2016.

Walker, M., Takayama, L., and Landay, J. **High-fidelity or low-fidelity, paper or computer médium?** Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting: HFES, p. 661-665, 2002.

Wheelmap. Disponível em: <<https://wheelmap.org/>>. Acesso: 17 mai. 2017.

WheelMate. Disponível em: <http://www.wheelmate.com/en/>. Acesso: 17 mai. 2017

WONG, W. C.; EYADAT, M. N. S. **Degree of freedom - experience of applying software framework.** In: PROCEEDINGS OF THE THIRD INTERNATIONAL CONFERENCE ON INFORMATION TECHNOLOGY: NEW GENERATIONS (ITNG'06), 2006. [s.n.], 2006.

W3C. Web Services Architecture: W3C Working Group 2004. Disponível em: <<https://www.w3.org/TR/ws-arch/>>. Acesso: 23 mai. 2017.

APÊNDICES

APÊNDICE A – CÓDIGO CONTROLLER.JS

```

angular.module('app.controllers', [])

//controle responsavel pela criacao de novas avaliacoes de locais já existentes
.controller('novaavaliacaoCtrl', function(
  $stateParams,
  LocalName,
  $cordovaToast,
  $scope,
  $cordovaCamera,
  $cordovaFileTransfer,
  $http,
  $cordovaImagePicker,
  $ionicPopup,
  $cordovaGeolocation,
  $ionicLoading)
{

  $scope.tipo = $stateParams.local_tipo;
  //sempre que a view relacionada a esse controle for vista, a funcao newLocalName() será
  chamada
  $scope.$on("$ionicView.enter", function(event, data){
  //handle event
  LocalName.newLocalName();

  });

  //tirar foto e realizar upload para o servidor
  $scope.takePhoto = function() {

  var local_id = $stateParams.local_id;
  var link = "http://rota-slz.esy.es/rota_slz/api/novaAvaliacao ";
  var id_usuario = window.localStorage.getItem("id_usuario");

  $ionicLoading.show();

```

```
//dependendo do tipo do local, as informacoes sao definidas
if($stateParams.local_tipo == 1){
var item =
{
calcada: $scope.calcada,
entradas: $scope.entradas,
veiculos: $scope.veiculos,
locomocao: $scope.locomocao,
banheiros: $scope.banheiros,
comentario: $scope.comentario,
local_id: $stateParams.local_id,
id_usuario: id_usuario,
nome_imagem: window.localStorage.getItem("nome_imagem"),
tipo: $stateParams.local_tipo
};
}

if ($stateParams.local_tipo == 2) {
var item = {
calcamento: $scope.calcamento,
autonomia: $scope.autonomia,
rampas: $scope.rampas,
comentario: $scope.comentario,
local_id: $stateParams.local_id,
id_usuario: id_usuario,
nome_imagem: window.localStorage.getItem("nome_imagem")
};
}

//requisao post com as informacoes da avaliacao de um novo local
$http.post(link, item).then(function (res){
console.log(res.data);
});
```

```

//opcoes da imagem
var options = {
fileKey: "avatar",
fileName: window.localStorage.getItem("nome_imagem"),
chunkedMode: "false",
mimeType: "false",
quality: 50,
destinationType: Camera.DestinationType.DATA_URL,
sourceType: Camera.PictureSourceType.CAMERA,
encodingType: Camera.EncodingType.JPEG,
                correctOrientation: true,
saveToPhotoAlbum: false,
};

//a camera é ativada
$cordovaCamera.getPicture(options).then(function(imageData) {
$scope.imgURI = "data:image/jpeg;base64," + imageData;
var targetPath = "data:image/jpeg;base64," + imageData;
$scope.targetPath = targetPath;

if ($scope.targetPath === undefined || $scope.targetPath === null) {
    $ionicPopup.alert({
    title: 'Error',
    template: 'Kindly upload file first'
    });
return false;
} else {

//buscar nome da imagem para ser salva no servidor

var option = { fileName: window.localStorage.getItem("nome_imagem") };

//upload da imagem é realizada, e a nova avaliacao é finalizada
$cordovaFileTransfer.upload(link, targetPath, option)
.then(function(result) {

```



```

$ionicLoading.hide();
$cordovaToast.show('Local salvo com sucesso!', 'short', 'bottom');
console.log(result);

}, function(err) {
$cordovaToast.show(err.message, 'short', 'bottom');
});
}

},function(err) {
//An error ocured. Show a message to the user
});
    };
})

//Controlador da tela do mapa
.controller('mapaCtrl', function($scope, $ionicLoading, $compile, $cordovaGeolocation, $http,
GoogleMaps) {

$scope.$on("$ionicView.enter", function(event, data){
//Iniciando a api do google maps
GoogleMaps.init();
});

})

//controle da tela de avaliação de um determinado local
.controller('avaliacaoCtrl', function($scope, $stateParams, $http) {

var local_id = $stateParams.local_id;
var item = {local_id: local_id};
var link = "http://rota-slz.esy.es/rota_slz/api/getComentarios";
$scope.listAvaliacao= [];

```

```

$scope.$on("$ionicView.enter", function(event, data){
//handle event
//enviando o id do local e buscando as suas informações
$http.post(link, item).then(function (res){
console.log(res.data);
$scope.res = res.data;

});

});

$scope.options = {
loop: false,
effect: 'fade',
speed: 500,
    }

$scope.$on("$ionicSlides.sliderInitialized", function(event, data){
//data.slider is the instance of Swiper
$scope.slider = data.slider;
});

})

//controle da tela de inserção de um novo local
.controller('novoLocalCtrl', function($state, LocalName, $cordovaToast, $scope,
$cordovaCamera, $cordovaFileTransfer, $http, $cordovaImagePicker, $ionicPopup,
$cordovaGeolocation, $ionicLoading) {

$scope.$on("$ionicView.enter", function(event, data){

//gerando o nome da imagem que será feita do novo local
LocalName.newLocalName();

//acessando o gps do dispositivo, e buscando as coordenadas atuais

```

```
var posOptions = {timeout: 10000, enableHighAccuracy: false};
$cordovaGeolocation
.getCurrentPosition(posOptions)
.then(function (position) {
lat = position.coords.latitude
long = position.coords.longitude
window.localStorage.setItem("latitude", lat);
window.localStorage.setItem("longitude", long);
$scope.latitude = lat;
$scope.longitude = long;
}, function(err) {
//error
});

});

//função que acessa a camera para tirar a foto
$scope.takePhoto = function() {
$ionicLoading.show();
    var link = "http://rota-slz.esy.es/rota_slz/api/novolocal";

var latitude = window.localStorage.getItem("latitude");
var longitude = window.localStorage.getItem("longitude");
var id_usuario = window.localStorage.getItem("id_usuario");

if($scope.tipo == 1){

var item = {
nome_imagem: window.localStorage.getItem("nome_imagem"),
id_usuario: id_usuario,
latitude: latitude,
longitude: longitude,
nome_local: $scope.nome_local,
tipo: $scope.tipo,
comentario: $scope.comentario,
```

```
calcada: $scope.calcada,
entradas: $scope.entradas,
veiculos: $scope.veiculos,
locomocao: $scope.locomocao,
banheiros: $scope.banheiros
};

}

if($scope.tipo == 2){

var item = {
nome_imagem: window.localStorage.getItem("nome_imagem"),
id_usuario: id_usuario,
latitude: latitude,
longitude: longitude,
nome_local: $scope.nome_local,
tipo: $scope.tipo,
comentario: $scope.comentario,
calcamento: $scope.calcamento,
autonomia: $scope.autonomia,
rampas: $scope.rampas
};

}

//enviando informações do novo local
$http.post(link, item).then(function (res){
console.log(res.data);
});

//configurando a nova imagem
var options = {
fileKey: "avatar",
fileName: window.localStorage.getItem("nome_imagem"),
```

```

chunkedMode: "false",
mimeType: "false",
quality: 50,
destinationType: Camera.DestinationType.DATA_URL,
sourceType: Camera.PictureSourceType.CAMERA,
encodingType: Camera.EncodingType.JPEG,
//popoverOptions: CameraPopoverOptions,
correctOrientation: true,
saveToPhotoAlbum: true
};

//inviando a nova imagem
$cordovaCamera.getPicture(options).then(function(imageData) {
$scope.imgURI = "data:image/jpeg;base64," + imageData;
var targetPath = "data:image/jpeg;base64," + imageData;
$scope.targetPath = targetPath;

if ($scope.targetPath === undefined || $scope.targetPath === null) {
    $ionicPopup.alert({
        title: 'Error',
        template: 'Kindly upload file first'
    });
    return false;
} else {
    var option = { fileName: window.localStorage.getItem("nome_imagem") };

    $cordovaFileTransfer.upload(link, targetPath, option)
        .then(function(result) {
            $ionicLoading.hide();
            $cordovaToast.show('Local salvo com sucesso!', 'short', 'bottom');
            $state.go('menu.mapa');

        }, function(err) {
            $cordovaToast.show(err.message, 'short', 'bottom');
        });
}
}

```

```

}

},function(err) {
//An error occurred. Show a message to the user
});

};

})

//controle da tela de perfil do usuário
.controller('perfilCtrl', function($scope, $cordovaGeolocation, $state, $http) {

//busca todas as informacoes do usuario no localStorage (armazenamento local), e os exibe em
"perfil.html"
$scope.id_usuario = window.localStorage.getItem("id_usuario");
$scope.nome = window.localStorage.getItem("nome");
$scope.email = window.localStorage.getItem("email");

var posOptions = {timeout: 10000, enableHighAccuracy: false};
$cordovaGeolocation
.getCurrentPosition(posOptions)
.then(function (position) {
var lat = position.coords.latitude
var long = position.coords.longitude
$scope.latitude = lat;
$scope.longitude = long;
                console.log("latitude: "+$scope.latitude+", longitude: "+$scope.longitude);
}, function(err) {
//error
});

//função de logout do aplicativo
$scope.sair = function(){

```

```

window.localStorage.removeItem("id_usuario");
window.localStorage.removeItem("nome");
window.localStorage.removeItem("email");

$state.go("entrar");
console.log("Saiu!");

}
})

//controle da tela de login
.controller('entrarCtrl', function($scope, $http, $ionicLoading, $state) {

$scope.entrar = function(){

var item = {email: $scope.email, password: $scope.senha};
var link = " http://rota-slz.esy.es/rota_slz/api/login";

$ionicLoading.show();
$http.post(link, item).then(function (res){
console.log(res.data.status);
if (res.data.status === "erro") {
$ionicLoading.hide();
alert("Email ou senha não encontrados");
}else{
window.localStorage.setItem("id_usuario", res.data.id);
window.localStorage.setItem("nome", res.data.nome);
window.localStorage.setItem("email", res.data.email);
$ionicLoading.hide();
$state.go('menu.mapa');
        }
});
};

})

```

```
//controle da tela de registro de novo usuário
.controller('registrarCtrl', function($scope, $http, $ionicLoading, $state) {

$scope.registrar = function(){

if($scope.nome!=null && $scope.email!=null && $scope.senha!=null){

var item = {nome: $scope.nome, email: $scope.email, senha: $scope.senha};
var link = "http://rota-slz.esy.es/rota_slz/api/registrar ";

$ionicLoading.show();
$http.post(link, item).then(function (res){
console.log(res.data);
$ionicLoading.hide();
$state.go('entrar');
});

}else{
alert("Todos os campos são obrigatorios!");
}

};

})
```


APÊNDICE B – IMPLEMENTAÇÃO API DO GOOGLE MAPS

```
//buscando informações dos locais e criando os marcadores
.factory('Markers', function($http) {

    var markers = [];

    return {
        getMarkers: function(){

            return $http.get("http://rota-slz.esy.es/rota_slz/api/getLocais").then(function(response){
                markers = response;
                return markers;
            });

        }
    }
})

.factory('GoogleMaps', function($cordovaGeolocation, Markers){

    var apiKey = false;
    var map = null;

    //inicia o mapa
    function initMap(){

        //opções do gps
        var options = {timeout: 10000, enableHighAccuracy: true};

        $cordovaGeolocation.getCurrentPosition(options).then(function(position){

            //latitude e longitude do local
            var latLng = new google.maps.LatLng(position.coords.latitude, position.coords.longitude);

            //opções do mapa
            var mapOptions = {
                center: latLng,
                zoom: 15,
                mapTypeId: google.maps.MapTypeId.ROADMAP
            };

            //novo objeto mapa
            map = new google.maps.Map(document.getElementById("map"), mapOptions);

            //aguarda até o mapa ser carregado
            google.maps.event.addListenerOnce(map, 'idle', function(){

                //carrega os marcadores
                loadMarkers();

            });

        });

    }
});
```

```

    }, function(error) {
        console.log("Could not get location");

        //Load the markers
        loadMarkers();
    });

}

//função que carrega os marcadores
function loadMarkers() {

    //busca todos os marcadores vindos do Markers factory
    Markers.getMarkers().then(function(markers) {

        console.log("Markers: ", markers);

        var records = markers.data;

        for (var i = 0; i < records.length; i++) {

            var record = records[i];
            var markerPos = new google.maps.LatLng(record.latitude, record.longitude);

            // Add o marcador no mapa
            var marker = new google.maps.Marker({
                map: map,
                animation: google.maps.Animation.DROP,
                position: markerPos,
                icon: 'http://rota-slz.esy.es/rota_slz/icons/splash.png'
            });

            //janela com a informação do local
            var infoWindowContent = "<h4>" + record.nome + "</h4><br><a href='#/avaliacao/'+"
            record.id + "' id='entrar-button2' class=" + " style='font-size: 24px;'>Ver
            Avaliações</a><br><br><br><a href='#/novaavaliacao/'+" record.id + "'>Nova Avaliação</a>";

            //chamando funcao que adiciona a janela de informacao na tela
            addInfoWindow(marker, infoWindowContent, record);

        }

    });

}

//funcao que adiciona a janela de informacao na tela
function addInfoWindow(marker, message, record) {

    var infoWindow = new google.maps.InfoWindow({

```

```
        content: message,  
        maxWidth: 900  
    });  
  
    google.maps.event.addListener(marker, 'click', function () {  
        infoWindow.open(map, marker);  
    });  
  
    }  
  
    return {  
        init: function(){  
            initMap();  
        }  
    }  
  
    })
```