

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS CORNÉLIO PROCÓPIO
BACHARELADO EM ENGENHARIA DE SOFTWARE

AMARI MARTINS JÚNIOR

**APLICATIVO MÓVEL COM FOCO NO TRAJETO DE ROTAS PARA
PORTADORES DE DEFICIÊNCIA MOTORA OU PESSOAS COM MOBILIDADE
REDUZIDA**

TRABALHO DE CONCLUSÃO DE CURSO

CORNÉLIO PROCÓPIO
2022

AMAURO MARTINS JÚNIOR

**APLICATIVO MÓVEL COM FOCO NO TRAJETO DE ROTAS PARA
PORTADORES DE DEFICIÊNCIA MOTORA OU PESSOAS COM MOBILIDADE
REDUZIDA**

Trabalho de Conclusão de Curso de
graduação, apresentado à disciplina
Trabalho de Conclusão de Curso, do curso
de Bacharelado em Engenharia de Software
da Universidade Tecnológica Federal do
Paraná – UTFPR, como requisito parcial
para a obtenção do título de Bacharel.

Orientador: Prof. Dr. Fabrício Martins Lopes

CORNÉLIO PROCÓPIO
2022

 4.0 International	Esta licença permite que outros remixem, adaptem e criem a partir do trabalho licenciado para fins não comerciais, desde que atribuam ao autor o devido crédito e que licenciem as novas criações sob termos idênticos.
---	--



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Câmpus Cornélio Procópio
Diretoria de Graduação e Formação Profissional
Departamento de Computação
Bacharelado em Engenharia de Software



TERMO DE APROVAÇÃO

Aplicativo móvel com foco no trajeto de rotas para portadores de deficiência motora ou pessoas com mobilidade reduzida

por

Amauri Martins Júnior

Este Trabalho de Conclusão de Curso de graduação foi julgado adequado para obtenção do Título de “Bacharel em Engenharia de Software” e aprovado em sua forma final pelo Programa de Graduação em Bacharelado em Engenharia de Software da Universidade Tecnológica Federal do Paraná.

Cornélio Procópio, 21/11/2022

Prof. Dr. Fabrício Martins Lopes

Prof. Dr. Francisco Pereira Junior

Prof. Dr. Henrique Shishido

“A Folha de Aprovação assinada encontra-se na Coordenação do Curso”

Dedico este trabalho única e exclusivamente à minha mãe, Rosa Maria Gomes da Silva, que sempre acreditou em mim e me apoiou independente das circunstâncias.

AGRADECIMENTOS

Agradeço ao meu orientador Prof. Dr. Fabrício Martins Lopes, que mesmo não sendo a sua área de pesquisa, aceitou me orientar e deu o seu melhor.

Agradeço a minha família por todo o suporte nesta jornada.

Por fim, agradeço a todos que de alguma maneira me ajudaram na realização deste projeto.

RESUMO

MARTINS, Amauri. **Aplicativo móvel com foco no trajeto de rotas para portadores de deficiência motora ou pessoas com mobilidade reduzida**. 2022. 53 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

É importante a melhoria da qualidade de vida da população e muitos esforços nas mais diversas áreas têm sido desenvolvidos. Um público em particular que merece atenção são os portadores de deficiência motora ou de mobilidade reduzida. Logo, é necessário o desenvolvimento de ações que contribuam para o auxílio na locomoção das pessoas com limitações motoras. Este trabalho tem como objetivo o desenvolvimento de uma solução de software para auxiliar tais pessoas a transitarem para seus destinos, informando-as se existe uma rota possível, qual a melhor para se seguir e qual a qualidade das ruas e calçadas que as mesmas passarão.

Palavras-chave: Aplicativo Móvel. Deficientes Motores. Rotas. Vias Públicas.

ABSTRACT

MARTINS, Amauri. **Mobile application with focus on routes for people with motor disabilities or people with reduced mobility**. 2022. 53 f. Trabalho de Conclusão de Curso (Graduação) – Engenharia de Software. Universidade Tecnológica Federal do Paraná. Cornélio Procópio, 2022.

It is important to improve the quality of life of the population and many efforts in various areas have been developed. One particular public that deserves attention is people with motor disabilities or reduced mobility. Therefore, it is necessary to develop actions that contribute to the aid in locomotion of people with this disability.. This work has the objective of proposing the development of a software solution to help such people to transit to their destinations, informing them if there is a possible route, which is the best to follow and what is the quality of the streets and sidewalks that they will pass.

Keywords: Mobile Application. Motor Disabilities. Routes. Public Roads.

Sumário

1 INTRODUÇÃO	9
1.1 JUSTIFICATIVA	10
1.2 OBJETIVOS	11
1.2.1 OBJETIVOS ESPECÍFICOS	12
1.3 ORGANIZAÇÃO DO TEXTO	12
2. FUNDAMENTAÇÃO TEÓRICA	13
2.1 TRABALHOS RELACIONADOS	13
3. DESENVOLVIMENTO	17
3.1 TECNOLOGIAS E FERRAMENTAS	17
3.2 MÉTODO	18
3.3 PROPOSTA DE DESENVOLVIMENTO	20
3.4 PROTÓTIPOS DE TELA	22
3.5 CRONOGRAMA	34
3.6 ARQUITETURA BACK-END	35
3.6.1 ARQUITETURA CLOUD	38
3.6.1.1 AZURE STORAGE ACCOUNT	38
3.6.1.2 AZURE FUNCTIONS	40
3.7 ARQUITETURA MOBILE	43
3.7.1 SINALIZAÇÃO DA ROTA NÃO ACESSÍVEL NO APLICATIVO	44
3.8 AUTENTICAÇÃO	45
3.8.1 LOGIN COM GOOGLE PELO APLICATIVO	46
3.8.2 AUTENTICAÇÃO NO BACK-END	47
3.8.3 JSON WEB TOKEN	48
3.9 DOCUMENTAÇÃO DA API	50
4. CONCLUSÕES	52
4.1 LIMITAÇÕES	52
4.2 TRABALHOS FUTUROS	52
4.3 CONSIDERAÇÕES FINAIS	53
REFERÊNCIAS	54

1 INTRODUÇÃO

No cenário atual, em meio a uma diversidade de tecnologias, surge a necessidade de se discutir sobre um tema demasiadamente relevante: a acessibilidade. De acordo com Nunes Leal (2012), a população brasileira corresponde a 190 milhões de pessoas, 7% desta população são portadoras de deficiência motora e no que diz respeito à mobilidade urbana, deve-se assegurar que tais pessoas tenham seus direitos preservados, conforme a legislação brasileira.

Segundo os artigos 1º e 46º do Estatuto da Pessoa com Deficiência, lei Nº 13146 de 6 de julho de 2015, está discriminado respectivamente que “Art. 1º É instituída a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência), destinada a assegurar e a promover, em condições de igualdade, o exercício dos direitos e das liberdades fundamentais por pessoa com deficiência, visando à sua inclusão social e cidadania.” e “Art. 46. O direito ao transporte e à mobilidade da pessoa com deficiência ou com mobilidade reduzida será assegurado em igualdade de oportunidades com as demais pessoas, por meio de identificação e de eliminação de todos os obstáculos e barreiras ao seu acesso.”, fazendo-se assim, essencial que as pessoas com ou sem deficiência ou mobilidade reduzida consigam se locomover por meio das vias públicas.

O portal Mobilize Brasil¹ é uma realização da Associação Abaporu, organização sem fins lucrativos fundada em 2003, e que atua nas áreas de educação, cultura e cidadania com o objetivo de contribuir com a melhoria da mobilidade urbana e da qualidade de vida nas cidades brasileiras. Essa associação realizou uma pesquisa em abril de 2012 em algumas capitais do Brasil sobre a qualidade das calçadas. Foi constatado que na maioria das capitais há um certo descaso em relação à conservação das calçadas especialmente por conta de obras frequentes realizadas por concessionárias de serviços públicos, sendo que, raramente após uma obra a via de passeio é restaurada conforme seu padrão de qualidade original (MOBILIZE, 2019).

¹ <http://www.mobilize.org.br/campanhas/calçadas-do-brasil/levantamento>

Atualmente existem aplicativos que fornecem algumas informações de locomoção, tais como tempo estimado de viagem e rotas mais rápidas como o Google Maps (Li e Zhijian, 2012) e o Waze (Google, 2019), aplicativos criados em 2005 e 2008 respectivamente, com a função de apresentar o mapa assim como informações sobre as rotas, porém na maioria dos casos é limitada à carros, ônibus, motos e bicicletas, não contendo as mesmas informações para quem faz uso de equipamentos para locomoção e, conseqüentemente, não é possível adquirir noção de qualidade das vias de passeio, que seria por onde o portador de deficiência motora transitaria.

Nesse contexto, esse trabalho realiza o desenvolvimento de um aplicativo destinado a pessoas com mobilidade reduzida, com a finalidade de ajudá-las a saber se seu trajeto é acessível, em Cornélio Procópio.

1.1 JUSTIFICATIVA

No cenário de acessibilidade atual, em que aproximadamente 13 milhões e 500 mil pessoas possuem alguma limitação motora (IBGE), foi identificado que há uma grande dificuldade em encontrar ferramentas voltadas ao auxílio de deficientes motores para com a sua locomoção que englobe tanto o tempo estimado de uma viagem até qual a melhor rota para a mesma.

Ao pesquisar utilizando as palavras chaves “aplicativo” e “deficiência motora”, é retornado diversos aplicativos que auxiliam outros tipos de deficiência, como deficiência auditiva e visual.

Nota-se que há uma grande variedade de aplicativos com a finalidade de auxiliar a locomoção, porém esses, em sua grande maioria, não possuem módulos específicos para a acessibilidade de pessoas com mobilidade reduzida.

Um exemplo da dificuldade que um deficiente motor pode sofrer é encontrado em um artigo publicado na Agência Brasil (Souza, 2019), no qual o indivíduo com deficiência física pontua sua dificuldade em se locomover pelas ruas de São Paulo, sendo que essas ruas muitas vezes não são acessíveis para facilitar a rota do mesmo.

A falta de aplicativos com essa funcionalidade afeta as pessoas que necessitam de algum tipo de equipamento para se locomover, causando insegurança, pois torna-se complicado a situação dos que não conhecem a qualidade da rua e da calçada por onde irão passar.

Com a escassez desse tipo de aplicativo, pode acarretar em viagens perdidas por chegar a um determinado ponto e se deparar com um caminho inviável, em ter que recorrer a meios de transportes individuais como os táxis, que, dependendo da distância, possui uma tarifa alta em relação aos transportes coletivos que possuem rotas preestabelecidas, ou na pior das hipóteses, pode haver algum acidente envolvendo o pedestre justamente pelo fato de não ter conhecimento prévio do caminho que irá transitar.

Portanto, o desenvolvimento do aplicativo se justifica por proporcionar informações de mobilidade que contribui com a segurança dos deficientes motores ou pessoas com mobilidade reduzida da cidade de Cornélio Procópio no Paraná, para que possam traçar a rota onde desejam chegar, podendo desta maneira, decidir se é viável ou não, diminuindo assim a possibilidade de ocorrer situações inconvenientes e ou prejudiciais.

1.2 OBJETIVOS

Este trabalho tem como objetivo geral o desenvolvimento de um aplicativo para smartphone capaz de auxiliar o usuário a visualizar o nível de qualidade das vias de passeio, mostrando características como rampas de acesso para cadeiras de rodas, piso tátil e nivelamento, tornando possível para o usuário uma tomada de decisão para qual caminho seguir, além de traçar a melhor rota e definir se é viável o trajeto, na cidade de Cornélio Procópio no Paraná.

1.2.1 OBJETIVOS ESPECÍFICOS

- Apresentar um mapa da cidade de Cornélio Procópio com suas ruas e calçadas mapeadas e categorizadas.
- Possibilitar o traço de rotas de um ponto A a um ponto B.
- Demonstrar de forma visual, com um estudo de psicologia das cores que será detalhado na seção 3.5, quais caminhos são viáveis para o trajeto de portadores de deficiência motora.
- Utilizar uma API fornecida pela Google que faça o cálculo por meio de estudo de grafos, qual a melhor rota a ser seguida.

1.3 ORGANIZAÇÃO DO TEXTO

Este trabalho encontra-se dividido em 4 capítulos, sendo que esse capítulo atual descreve o problema e aponta de maneira geral alguns dados que contribuem para a motivação de realizar o trabalho, além de detalhar quais os objetivos do trabalho.

No capítulo 2, se encontra a fundamentação teórica que apresentará os aplicativos relacionados que o projeto atual será baseado.

O capítulo 3 aborda o desenvolvimento do trabalho, descrevendo quais os diferenciais, as tecnologias e ferramentas que serão utilizadas, a metodologia de desenvolvimento do projeto, os protótipos de tela e, por fim, o cronograma.

Por fim, no capítulo 4, constam as considerações finais.

2. FUNDAMENTAÇÃO TEÓRICA

Como foi descrito no capítulo 1, existem alguns aplicativos que apresentam a funcionalidade de traçar o caminho e calcular quais as melhores rotas, sendo tanto para pedestres, quem está de carro, ônibus ou bicicleta.

2.1 TRABALHOS RELACIONADOS

Como descrito por Li e Zhijian (2012), a empresa Google apresentou em 2005 um serviço gratuito de mapas, conhecido como Google Maps. Esse serviço possui funcionalidades capazes de apresentar imagens dos mapas que podem ser tiradas tanto de drones quanto de satélites, além de apresentar informações de tráfego e rotas possíveis e também o tempo estimado para o deslocamento do ponto de partida ao ponto de chegada.

A Figura 1 exhibe de maneira efetiva imagens tiradas a partir de drones dos locais.

Atualmente o Google Maps conta com parcerias de órgãos públicos com a finalidade de mapear os trajetos feitos por ônibus, trens e metrô. Oferece também oportunidade das próprias pessoas contribuírem, abrindo programas para fotógrafos profissionais, viajantes e organizações, como câmaras de turismo, ONGs, agências governamentais, universidades ou grupos de pesquisa, para que tirem as fotos e compartilhem a experiência no serviço.

A empresa está realizando também parcerias com órgãos privados, para que além de apenas mapa, seja possível adicionar a localização de um determinado lugar, possibilitando que na hora de traçar as rotas, ao invés de informar o nome das ruas do ponto de partida e destino, possa ser informado também os locais, facilitando assim o manuseio da aplicação.



Figura 1 - Universidade Tecnológica Federal do Paraná no Google Maps.

Fonte: Google Maps.

De acordo com Google (2019), a API do Google Maps é uma tecnologia baseada em Javascript que é disponibilizada pela Google que providencia muitos serviços de processamento de mapas e conteúdos adicionais, também permite que os mapas sejam customizados e em sua grande maioria é utilizada por usuários desenvolvedores de aplicações web para incorporar os mapas em suas web pages.

A Figura 2 exhibe de maneira ilustrativa como funciona o traço de rotas e apresenta de maneira visual o tempo do percurso.

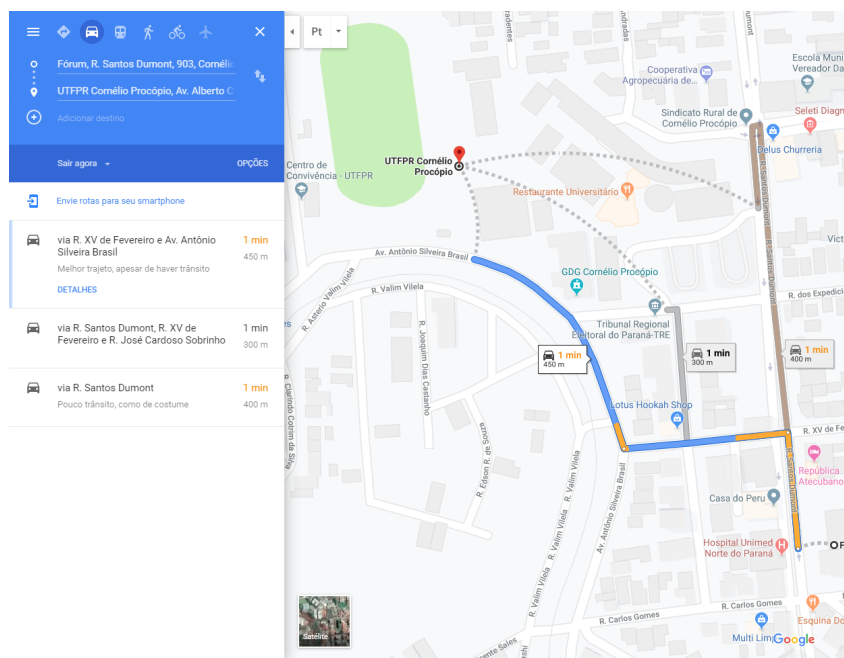


Figura 2: Rota do Fórum de Cornélio Procopio à UTFPR pelo Google Maps.
Fonte: Google Maps.

Existe também no contexto de aplicativos que funcionam na base da colaboração dos usuários, o Onde Tem Tiroteio. A OTT do Brasil (2019), criadora do aplicativo que foi feito com a missão de salvar os cidadãos de balas perdidas, arrastões e de falsas blitz, foi fundado em 2016 por quatro amigos que se preocupavam muito com o crescimento descontrolado da violência no Brasil.

Esse aplicativo se baseia na segurança feita de cidadão para cidadão (C2C), sendo que cada cidadão pode atualizar em tempo real a segurança ao seu redor, ajudando a ele e todos os outros participantes da rede de informações. Após os usuários enviarem os alertas tanto pelo aplicativo, quanto pelas redes sociais, as informações são checadas pela equipe operacional e após os alertas serem confirmados, eles são finalmente enviados para o aplicativo. Onde Tem Tiroteio funciona 24 horas por dia e 365 dias por ano, no momento está disponível nos estados do Rio de Janeiro (onde foi inicialmente criado) e agora também, São Paulo, com planos para que brevemente seja incorporado por mais estados do Brasil. A Figura 3 exibe um exemplo do aplicativo na sua tela de ocorrências.



**Figura 3: Tela de Alertas mais próximos da localidade de Cornélio Procópio.
Fonte: Onde Tem Tiroteio.**

3. DESENVOLVIMENTO

É proposto neste trabalho a implementação de um aplicativo *mobile* que possa contribuir e ser um grande aliado na rotina de pessoas com mobilidade reduzida, permitindo que as mesmas tracem rotas especificando seu ponto de partida e destino, para pessoas com mobilidade reduzida. O diferencial apresentado é que os usuários obterão as informações necessárias tais como se é possível chegar ao local escolhido, melhor rota e qualidade das ruas e calçadas por onde irá passar.

Inicialmente, o projeto foi pensado com o intuito de ajudar as pessoas de Cornélio Procópio no Paraná, que precisam de auxílio de equipamentos para se locomoverem, porém será feito de maneira com que os usuários consigam contribuir com os dados, podendo ser expandido para demais cidades.

3.1 TECNOLOGIAS E FERRAMENTAS

Para este trabalho será necessário o uso de ferramentas computacionais que auxiliarão na implementação do projeto, armazenamento dos dados, versionamento do código e gestão do projeto, as quais são resumidamente descritas na Tabela 1.

Tabela 1 - Ferramentas e Tecnologias utilizadas no desenvolvimento do projeto.

Atividade	Ferramenta
Gerenciamento de projeto	Trello
Banco de Dados	SQL Server
Desenvolvimento	Visual Studio Code
Repositório	Github
Linguagem de Programação Back End	C#
Linguagem de Programação Front End	React Native
Documento	Word 2019

Fonte: Autoria Própria, 2019.

3.2 MÉTODO

Como metodologia de desenvolvimento optou-se por adotar duas metodologias ágeis por ser o mais próximo das metodologias utilizadas no mercado de trabalho atual. É uma combinação de Scrum com Kanban como referido na obra de Terlecka (2012). De acordo com Sriastava, Bhardwaj e Saraswat (2017), o Scrum é uma metodologia ágil que ficou conhecida após sua incorporação na indústria automobilística para melhorar o processo de trabalho. Alguns anos depois, foi observado que não necessariamente funcionaria apenas para esse ramo, mas sim, para qualquer outro ramo e hoje, é bastante utilizado pelas empresas de desenvolvimento de software.

O Scrum conta com uma *timebox* e itens divididos em *backlogs* que ajuda na compreensão e organização do trabalho. É definido um tempo para a *timebox* e por meio das estimativas é possível estipular a quantidade de *backlogs* que devem ser entregues até o final da Sprint como evidenciado por Sriastava, Bhardwaj e Saraswat (2017). Essa etapa é conhecida como *Sprint Planning Meeting*. Ao final da *Sprint* é feito uma *Sprint Review* e também uma *Sprint Retrospective*, na qual é analisada a *Sprint* e funciona como um processo de evolução constante, em que é discutido pontos negativos, positivos e ações de melhoria. Durante o tempo da *Sprint*, é realizada a *Daily Meeting* que são as reuniões diárias para acompanhar o progresso da equipe em relação às entregas da *Sprint*. Assim que acaba uma *Sprint*, todo o processo é reiniciado.

Por outro lado, de acordo Matharu (2015), o Kanban é uma metodologia ágil que também está sendo bastante utilizada no mercado de trabalho e que possibilita visualizar todo o progresso existente no projeto por meio do Board, além de ser flexível em suas etapas. O Kanban é muito mais objetivo, basicamente consiste em uma fila de itens priorizados e prontos para serem desenvolvidos. O ideal do Kanban seria uma quantidade predefinida de itens para serem desenvolvidos e trabalhando com o conceito de WiP (*Work in Progress*) que é o máximo de trabalho

que sua equipe consegue desenvolver simultaneamente, isso acaba ajudando bastante a encontrar algumas possíveis falhas do processo.

Portanto foi escolhido essa adaptação do Scrum com o Kanban devido a seriedade e responsabilidades dos prazos do Scrum e o dinamismo e boa visualização do progresso do Kanban. Os *backlogs* são organizados no quadro de projetos do Trello com o formato de Kanban básico, com a prioridade definida por etiquetas que podem ser amarelas para prioridade baixa, laranja para prioridade média e vermelha para prioridade alta. Para definir se o *backlog* foi concluído, também é utilizado uma etiqueta, esta por sua vez com a cor verde, sinalizando que aquele *backlog* já está pronto.

A Figura 4 apresenta o quadro do projeto desenvolvido:

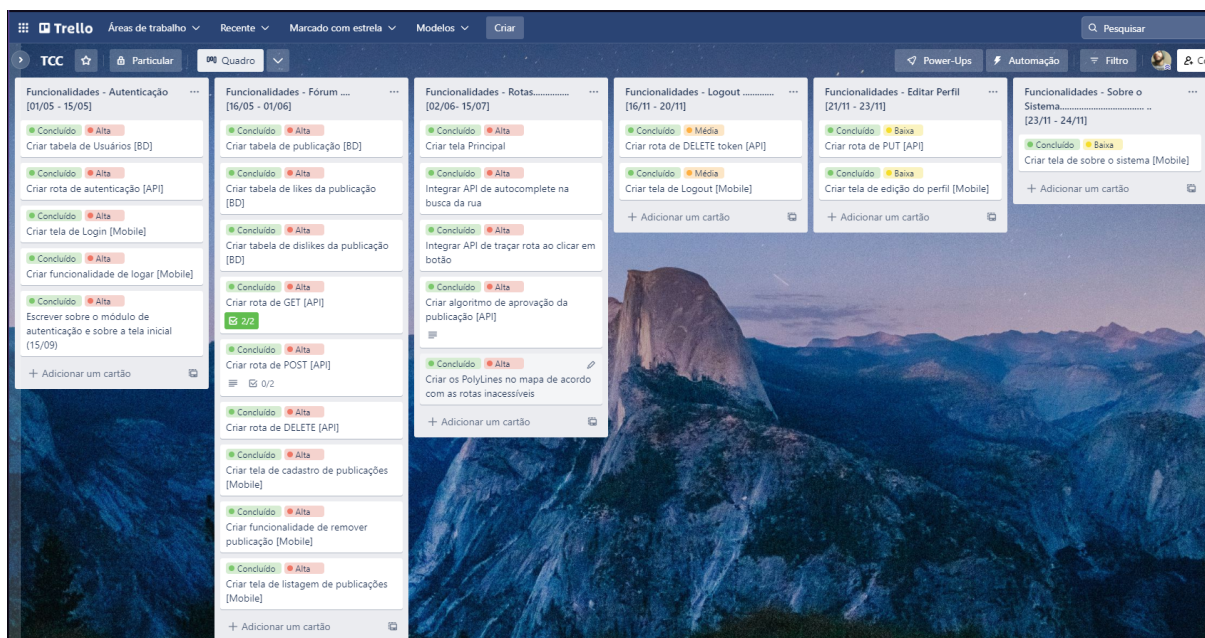


Figura 4: Quadro do projeto no Trello.

Fonte: Autoria própria, 2022.

Como apresentado na Figura 4, no início de cada coluna é descrito o módulo do sistema, a data inicial da *sprint* e também o prazo final para a conclusão das tarefas, sendo as etiquetas que categorizam as prioridades a serem seguidas dentro de cada módulo, de cada *sprint*.

3.3 PROPOSTA DE DESENVOLVIMENTO

O aplicativo é proposto para funcionar de maneira contributiva. Inicialmente é realizado um trabalho manual de mapeamento das calçadas de Cornélio Procopio, onde por meio de um *checklist* que pode ser encontrado na Figura 6, é definido se a calçada é acessível ou não. A ideia é que isso se torne algo escalável e que as pessoas consigam ajudar de maneira simples, basta seguir o *checklist* pelas calçadas que forem passando e reportando a avaliação direto pelo aplicativo, que funcionará por meio de votos que serão justificados por fotos.

Então, se algum usuário quiser reportar que a calçada é acessível, ele deverá ir no fórum do aplicativo e apresentar as fotos da calçada, contendo a rampa, nivelamento e qualidade da mesma. Essa publicação será então disponibilizada para que outros usuários consigam votar e os que votarem negativamente, também poderão justificar com fotos o porquê do seu voto. Ao final de um dia após a publicação, será analisada e se a mesma possuir ao menos 70% a mais de votos positivos sobre os negativos, será publicado o resultado como calçada acessível. O mesmo método serve para caso o usuário queira reportar uma calçada não acessível. O aplicativo também permitirá que se crie novas votações para as mesmas calçadas, uma vez que elas poderão sofrer alterações com o tempo. Pode ser conferido na Figura 5 como é o fluxo de reporte da calçada pelo aplicativo.

Reportar qualidade da calçada

AMAURI MARTINS JUNIOR | October 18, 2019

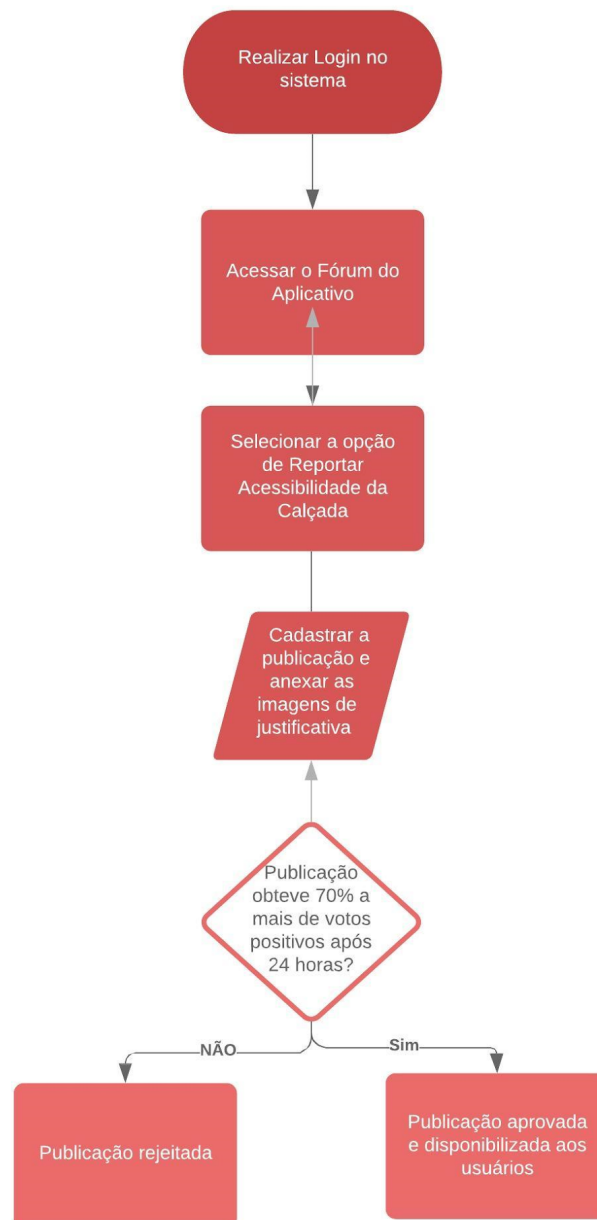
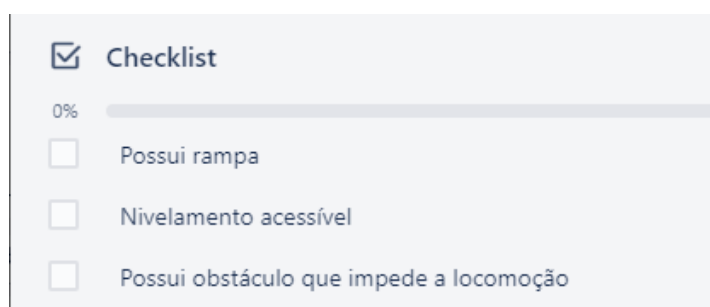


Figura 5: Fluxo de reporte da calçada.

Fonte: Autoria Própria, 2019.



A screenshot of a checklist titled 'Checklist' with a checked box icon. Below the title is a progress bar showing 0%. The checklist contains three items, each with an unchecked checkbox: 'Possui rampa', 'Nivelamento acessível', and 'Possui obstáculo que impede a locomoção'.

Item	Status
Possui rampa	0%
Nivelamento acessível	
Possui obstáculo que impede a locomoção	

Figura 6: *Checklist* de reporte da calçada.
Fonte: Autoria Própria, 2019.

3.4 PROTÓTIPOS DE TELA

Nesta seção são apresentados os protótipos iniciais das telas principais do sistema e é feita a comparação com a versão final do aplicativo e suas funcionalidades.



A screenshot of a mobile app registration screen. The background is purple. At the top, there is a pink header bar with a user icon in the center. Below the header, the text 'Rotas Acessíveis' is displayed in white. There are four white input fields stacked vertically, labeled 'Email', 'Nome Completo', 'Senha', and 'Confirmar Senha'. Below these fields is a pink button labeled 'Cadastrar'. At the bottom of the screen, the word 'Cadastrar' is written in white.

Figura 7: Protótipo da tela de cadastro do aplicativo.
Fonte: Autoria Própria, 2019



Figura 8: Protótipo da tela de login do aplicativo.
Fonte: Autoria Própria, 2019



Figura 9: Tela atual de login do aplicativo.
Fonte: Autoria Própria, 2022

A Figura 9 representa a tela final de login do aplicativo. Inicialmente foi proposto no protótipo, um cadastro e gerenciamento dos usuários no próprio aplicativo, como pode ser observado nas Figuras 7 e 8, mas para a versão final optou-se por utilizar o login social com a Google, que será descrito em detalhes na seção 3.8. Assim, quando o usuário clicar no botão de Entrar com o Google, ele deverá escolher o email e uma vez escolhido, o sistema iniciará a autenticação.

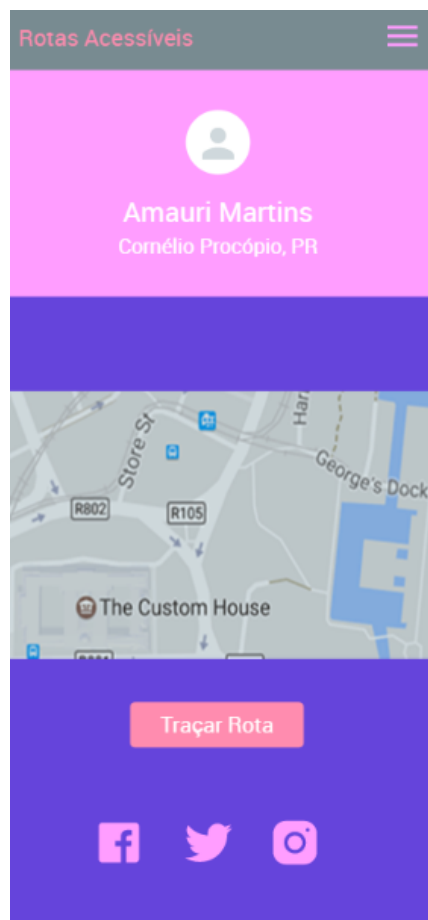


Figura 10: Protótipo da tela principal do aplicativo.
Fonte: Autoria Própria, 2019

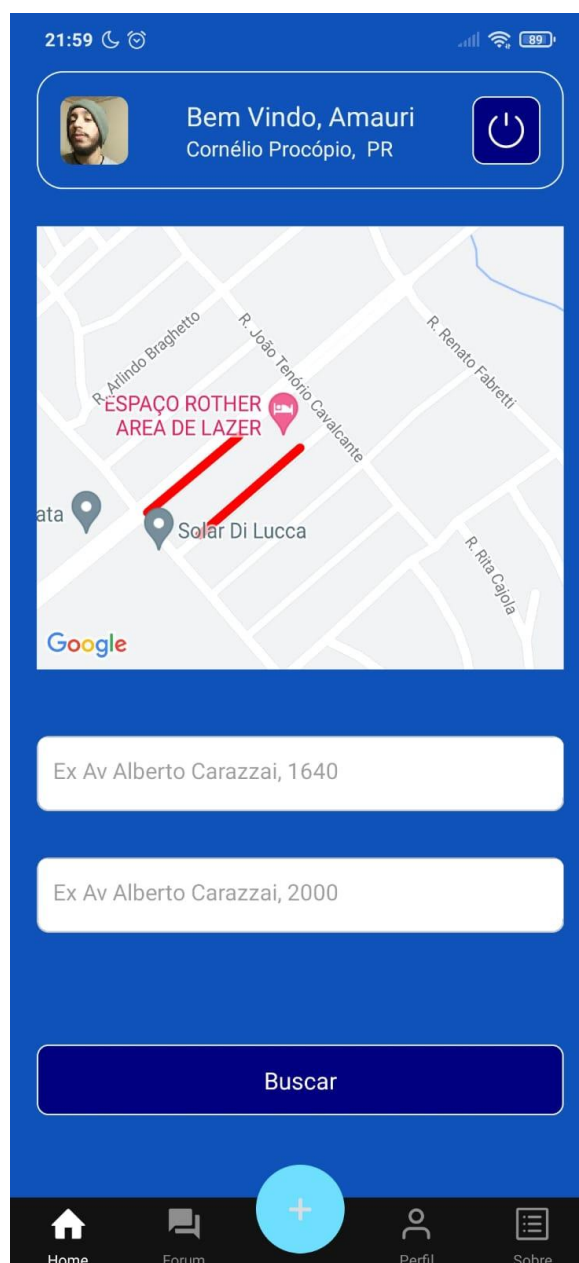


Figura 11: Tela atual de início do aplicativo.
Fonte: Autoria Própria, 2022

A Figura 11 apresenta a tela principal do aplicativo que é exibida logo após a autenticação. Nesta tela o usuário poderá encontrar o mapa da cidade e sinalizado em vermelho, encontrará os trechos que foram marcados como inacessíveis. Além disso, o usuário também poderá traçar uma rota da sua origem até o destino, com isso ele poderá observar se o trecho que terá que perseguir, passará por alguma rota inacessível. Como apresentado na Figura 12:

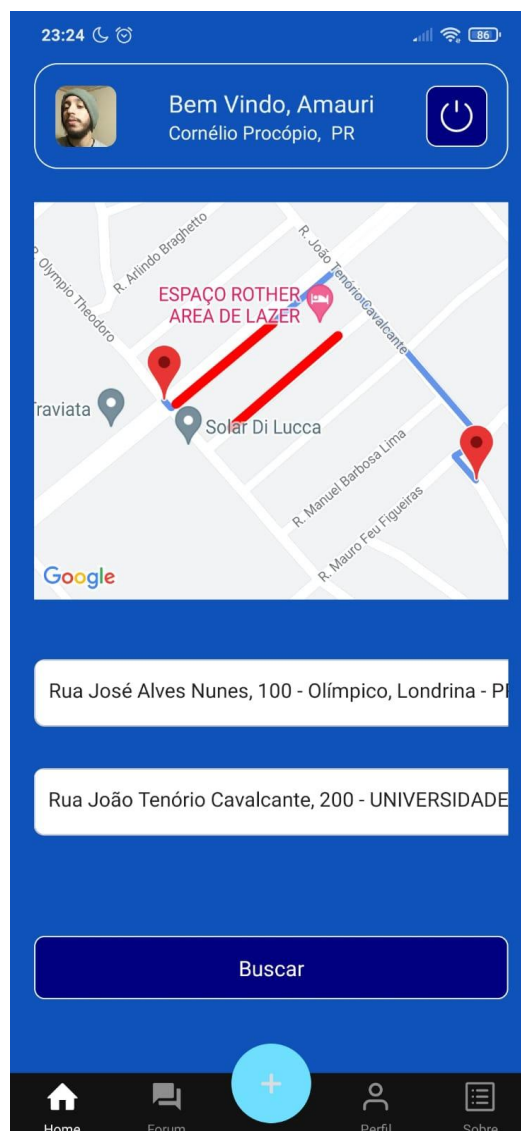


Figura 12: Tela atual de início do aplicativo com rota traçada passando por caminho inacessível.

Fonte: Autoria Própria, 2022

A Figura 12 exibe a rota traçada a partir da rua José Alves Nunes número 100 até a rua João Tenório Cavalcante número 200. O trecho em questão é sinalizado pela cor azul e passa por uma rota inacessível, que é sinalizada pela cor vermelha. Desta maneira o usuário poderá tomar a decisão de qual é o melhor caminho que deverá seguir.



Figura 13: Protótipo da tela de menu do aplicativo.
Fonte: Autoria Própria, 2019

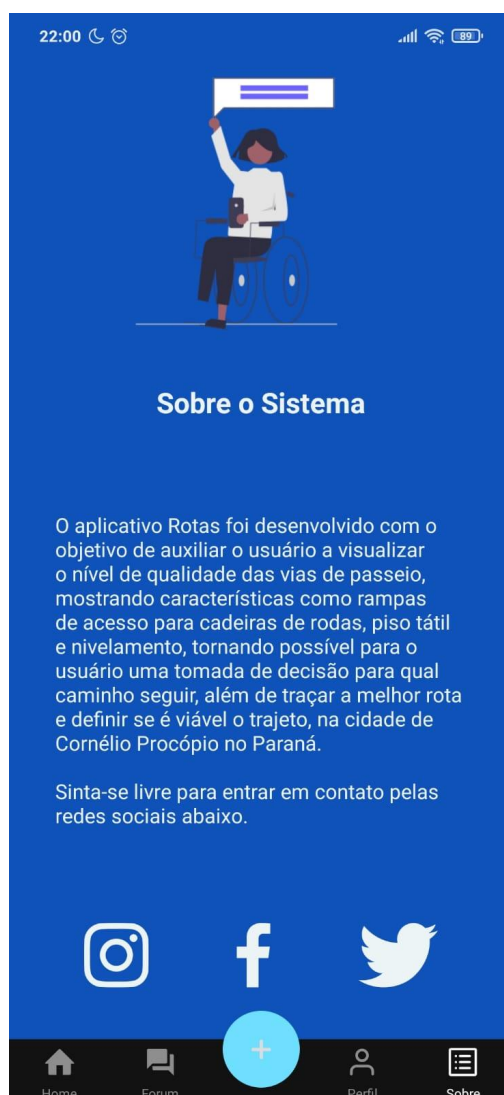


Figura 14: Tela atual de sobre o sistema do aplicativo.

Fonte: Autoria Própria, 2022

Na Figura 13 foi proposto um menu lateral onde o usuário poderia acessar as outras telas do aplicativo com o intuito de facilitar a experiência do usuário, optou-se pelo menu inferior, que pode ser observado na Figura 14, onde o usuário precisará clicar apenas uma vez no aplicativo para chegar na tela desejada, além de deixar o aplicativo com um aspecto moderno. Esta tela descreve o objetivo do aplicativo e oferece as opções para o usuário contatar o administrador pelas redes sociais.

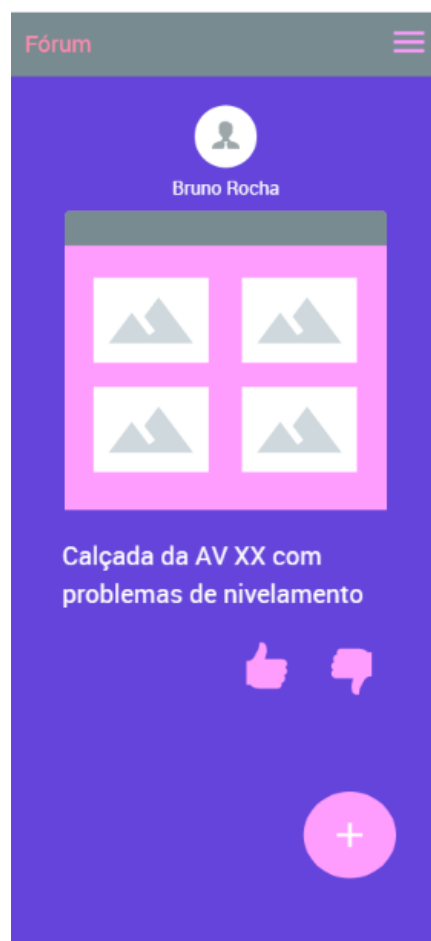


Figura 15: Protótipo da tela do fórum do aplicativo.
Fonte: Autoria Própria, 2019

21:59

Novo Post

Bairro

Digite o Bairro

A partir de

Ex Av Alberto Carazzai, 1640

Até

Ex Av Alberto Carazzai, 2000

Descrição

Ex Buraco na calçada

☐ Possui Rampa

☐ Nivelamento Acessível

☐ Possui Obstáculo

Home Forum Perfil Sobre

Figura 16: Tela atual de criação de post do aplicativo parte superior.
Fonte: Autoria Própria, 2022

A Figura 16 exibe a parte superior da tela de *Novo Post*, que é onde o usuário preencherá as informações necessárias para marcar uma rua como inacessível, ele poderá definir o bairro, o ponto de origem e destino, também poderá escrever uma breve descrição do problema encontrado na acessibilidade do local e, por fim, preencher o *checklist* da calçada, como foi descrito na Figura 6.

21:59 89%

Descrição

Ex Buraco na calçada

☐ Possui Rampa

☐ Nivelamento Acessível

☐ Possui Obstáculo

Imagens

Salvar

Home Forum Perfil Sobre

Figura 17: Tela atual de criação de post do aplicativo parte inferior.

Fonte: Autoria Própria, 2022

Na parte inferior da tela de criação de *post*, como mostra a Figura 17, o usuário tem a opção de selecionar até 4 imagens que ajudarão a comprovar a veracidade da publicação, caso o usuário não inclua nenhuma imagem, a publicação não será aceita.

22:00

Editar Perfil



Nome

Amauri Martins

Email

amaurij@alunos.utfpr.edu.br

CPF

Digite seu CPF

Cidade

Cornélio Procópio

UF

Home Forum Perfil Sobre

Figura 18: Tela atual de edição do perfil do aplicativo.
Fonte: Autoria Própria, 2022

Na Figura 18, pode-se observar a tela de edição do perfil do usuário. Os campos de nome, *email* e a foto de perfil são obtidos pela autenticação no Google. O usuário poderá também adicionar a cidade e a UF em que reside, estes últimos serão utilizados na tela inicial, na saudação ao usuário como pode ser visto na Figura 12.

3.5 CRONOGRAMA

Tabela 2 - Cronograma do desenvolvimento do projeto.

Atividade	Data de início	Data de fim
Pesquisar sobre acessibilidade e UX	10/07/2019	10/08/2019
Aprimorar conhecimentos em React Native	11/08/2019	01/09/2019
Elicitar os requisitos e dividir em backlogs a serem desenvolvidos	02/09/2019	02/10/2019
Criar as interfaces do aplicativo	03/10/2019	17/10/2019
Desenvolvimento da camada de Modelo	18/10/2019	01/12/2019
Desenvolvimento da Camada de Visualização	02/12/2019	15/01/2020
Desenvolvimento da Camada de Controle	16/01/2020	01/03/2020
Realizar os testes de cálculos de rota e eventuais correções	02/03/2020	02/04/2020
Elaboração do documento final	03/04/2020	15/05/2020
Apresentação do trabalho	01/06/2020	01/06/2020

Fonte: **Autoria Própria, 2019.**

Com o objetivo de criar uma interface elegante aos olhos e fácil de utilizar, fez-se necessário o estudo sobre acessibilidade e *UX* (Experiência do Usuário). Visto que se trata de um aplicativo voltado para pessoas com limitações motoras, após um estudo sobre a psicologia das cores voltadas para acessibilidade (PREFEITURA DE SÃO PAULO), foi escolhido o azul como cor predominante, uma vez que esta é a cor sempre utilizada para sinalizar locais exclusivos para pessoas com deficiência e também, segundo a Prefeitura de São Paulo (2019), essa é a cor

utilizada no símbolo internacional de acesso, desenhado pela Unidade de Desenho Gráfico do Departamento de Informação Pública das Nações Unidas, em Nova York, a pedido da Divisão de Reuniões e Publicações do Departamento de Assembléia Geral e Gestão de Conferências das Nações Unidas.

Para sinalizar as ruas inacessíveis, também utilizando a psicologia das cores, foi utilizada a cor vermelha, que remete perigo ao cérebro (GALLARDO, 2019).

Como linguagem de programação para desenvolvimento *mobile*, foi escolhido o React Native. Uma das atividades é aprimorar os conhecimentos técnicos nesta linguagem, tais como armazenamento de estado dos componentes, compartilhamento de parâmetros entre as telas, integração com API de terceiros e por fim, biblioteca de componentes, uma vez que esses pilares são fundamentais para o desenvolvimento do aplicativo proposto.

Para realizar as atividades de desenvolvimento das camadas de modelo, visualização e controle, foi escolhido o C#. A seção 3.6 descreve em detalhes o processo de desenvolvimento e também a arquitetura utilizada nesta etapa.

Devido à pandemia, o cronograma foi prorrogado em 2 anos e seis meses, mas a ordem das atividades foi mantida.

3.6 ARQUITETURA BACK-END

Utiliza-se como arquitetura de desenvolvimento do *back-end*, uma arquitetura em 4 camadas.

A primeira camada é a camada de Web, responsável por gerenciar os *Controllers* da aplicação, que são os *endpoints* da API, ou seja, quando o usuário fizer uma requisição, os *Controllers* serão os responsáveis por executar a ação que o usuário está requisitando. Nesta camada, também é definida as configurações como conexão de banco de dados, início da aplicação, chaves secretas e documentação da API.

A segunda camada é a *Application*, que gerencia a regra de negócio, nesta camada é onde são feitas as validações para definir se a requisição do usuário é válida ou não, caso a requisição seja válida, esta camada deverá prosseguir com a ação do usuário. A camada de aplicação também é responsável por definir os *Data Transfer Objects*, popularmente conhecido como DTO, que de acordo com DEVMEDIA (2013) é um objeto simples usado para transferir dados de um local a outro na aplicação, sem lógica de negócios em seus objetos. Pode-se entender o DTO como uma maneira de modificar ou adequar um objeto para a sua finalidade, por exemplo, em uma listagem dos seus amigos do Facebook, o usuário não deve ser capaz de poder ver a senha de login do amigo, isso porque na listagem de amigos do Facebook, existe um DTO para mostrar apenas as informações relevantes para aquele contexto.

Já a terceira camada, é a de *Domain*, que cuida do domínio da aplicação. Nesta camada é onde ficam as entidades do projeto. No cenário atual, tem-se a entidade de Usuário e a entidade de *Posts*, assim como uma entidade base que têm as propriedades em comum que todas as entidades terão.

As propriedades que toda entidade têm em comum é um código de identificação, representado pela propriedade *Id*, a data de criação da entidade, representado pela propriedade *CreatedOn*, assim como a data de atualização desta entidade que é retratado pela propriedade *UpdatedOn* e, por fim, a propriedade *IsActive*, que define se a entidade está ativa ou inativa, pois não utilizamos a remoção de registros do sistema, mas sim, desativamos.

Também na camada de domínio, é definido as interfaces de repositório, que contém os principais métodos de banco de dados para as entidades, tais como *Select*, onde é feito uma busca no banco de dados de acordo com a entidade que foi selecionada. *Update* que é quando é dada uma instrução de atualização de algum registro de alguma entidade e o *Insert*, que é quando é necessário inserir um novo registro em uma entidade, como por exemplo, um novo usuário do sistema.

Como descrito anteriormente, não é utilizado a instrução de *Delete* nas entidades, já que não é o ideal remover do banco de dados registros sensíveis.

Por último, tem-se a camada de Infra, que é dividida em 2 sub camadas: a camada de *Data*, que coordena o banco de dados do sistema e realiza o mapeamento das entidades para serem usadas como tabelas no banco de dados, mantém o histórico de modificações na estrutura do banco de dados e que também realiza a implementação das interfaces que foram definidas na camada de *Domain*.

A segunda subcamada é a de *CrossCutting*, que conduz o compartilhamento de um dado específico com toda a aplicação. No caso do projeto atual, é utilizada esta camada para realizar a configuração da Autenticação no sistema, tal configuração pode ser compartilhada em qualquer nível da aplicação e será detalhada no capítulo de Autenticação.

A Figura 19 ilustra de maneira visual como a arquitetura está dividida:

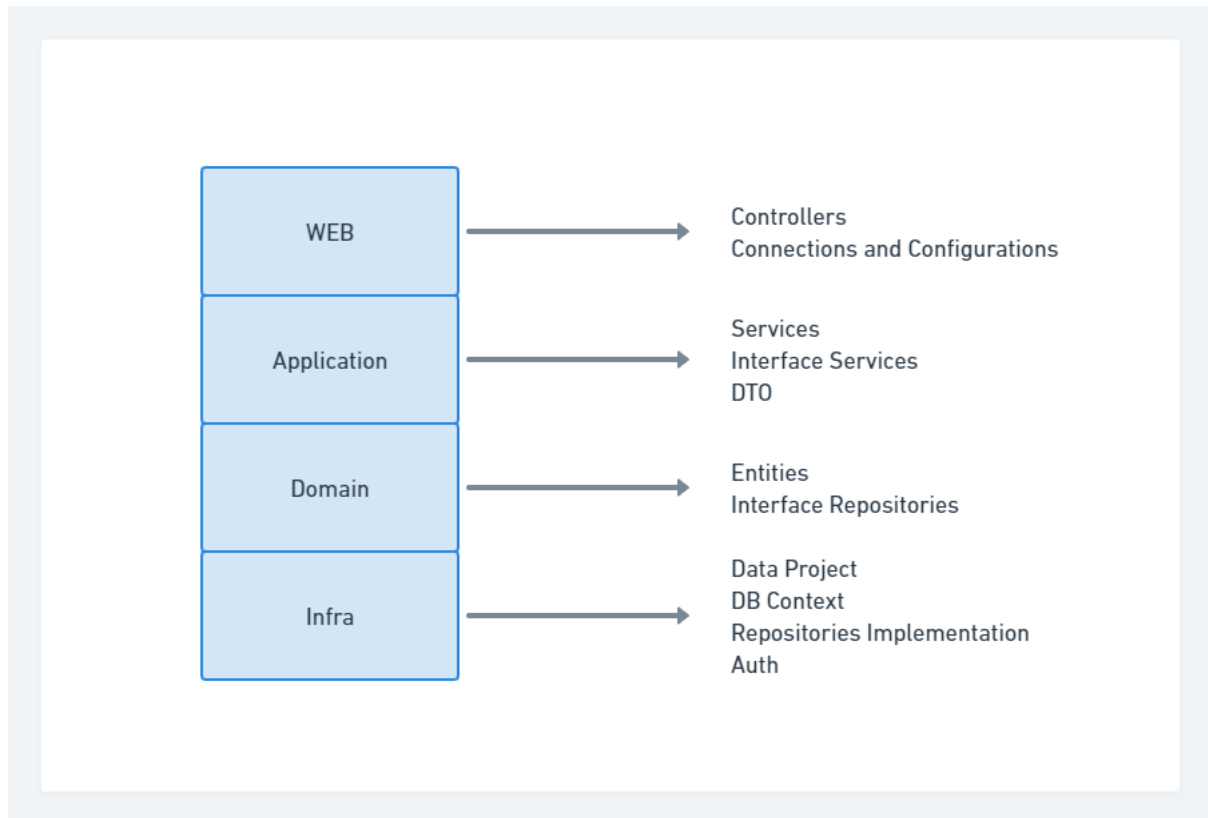


Figura 19: Arquitetura back-end.
Fonte: Autoria Própria, 2022

3.6.1 ARQUITETURA CLOUD

Para o armazenamento de imagens e o algoritmo de publicação ou rejeição do Post, foram utilizados os serviços da Azure na nuvem, sendo eles o Azure Storage (MICROSOFT) para o armazenamento e o Azure Functions (MICROSOFT) para executar o algoritmo periodicamente.

3.6.1.1 AZURE STORAGE ACCOUNT

O Azure Storage Account, ou também conhecido como Azure Blob, segundo a Microsoft (2022), armazena os objetos de dados, incluindo blobs que são objetos binários, além de disponibilizar o compartilhamento de arquivos, filas, tabelas e disco. Uma conta de armazenamento do Azure fornece um endereço exclusivo para os dados armazenados, que podem ser acessados de qualquer lugar do mundo por

conexão HTTP ou HTTPS. A Microsoft provê a segurança, alta disponibilidade e o escalonamento, caso necessário, dos arquivos armazenados.

Para implementar o armazenamento pelo Storage Account, foi criada uma conta no Azure e também criado um container dentro do Storage Account, que ficará responsável por agrupar os arquivos do sistema. Com o container criado, é necessário obter as credenciais de conexão e configurar no sistema parecido com a conexão do banco de dados.

Uma vez configurada a conexão para o container, basta transformar a imagem (que é recebida em formato base64 pelo aplicativo mobile) em um array de bytes e em seguida, transformar em um objeto do tipo Memory Stream, que armazenará esse array de bytes na memória. Tendo a imagem armazenada na memória, basta fazer o envio da imagem para o Azure, como mostra a Figura 20:



```
public string UploadImage(string base64Image)
{
    var filename = Guid.NewGuid().ToString() + ".jpg";

    var imageFormatted = new Regex(@"^data:image\/[a-z]+;base64,").Replace(base64Image, "");

    byte[] image = Convert.FromBase64String(imageFormatted);

    var blobClient = new BlobClient(_configuration.GetSection("Azure").GetValue<string>
("BlobConnectionString"), _configuration.GetSection("Azure").GetValue<string>("BlobContainer"),
filename);

    using (var stream = new MemoryStream(image))
    {
        blobClient.Upload(stream);
    }

    return blobClient.Uri.AbsoluteUri;
}
```

Figura 20: Envio de imagem para o Azure.

Fonte: Autoria Própria, 2022

Na Figura 20, o método *UploadImage* recebe a imagem no formato *base64* e por meio da interface *Guid*, gera um ID único para a imagem. Com isso, após transformar a imagem em um array de bytes, é utilizado o objeto *BlobClient*, que é disponibilizado pela Azure, por meio de uma biblioteca de objetos e funções que

podem ser utilizadas para usar os serviços da nuvem, para que seja possível através deste objeto ter acesso ao nosso container.

Para acessar o container, é necessário passar as configurações da chave de acesso e o nome do container na instanciação do objeto, assim como o nome do arquivo que deve ser único. Com o objeto criado, basta transformar o array de bytes em um Memory Stream, fazendo com que a imagem seja armazenada na memória da aplicação, e então utilizar o método Upload da biblioteca da Azure, passando a imagem.

Finalizando o envio para a Azure, é retornado o nome do arquivo único que foi criado previamente para que seja salvo no banco de dados. Esse nome será o endereço que será utilizado para acessar a imagem no Storage Account quando necessário.

Ao utilizar esta abordagem, delega-se a responsabilidade de armazenamento do arquivo para o serviço da Microsoft, e a única coisa que deve ser salvo é um texto representando o endereço do arquivo, que ocupa menos espaço no banco de dados do que um arquivo binário.

A Figura 21 exibe o fluxo do envio de imagem desde o início:

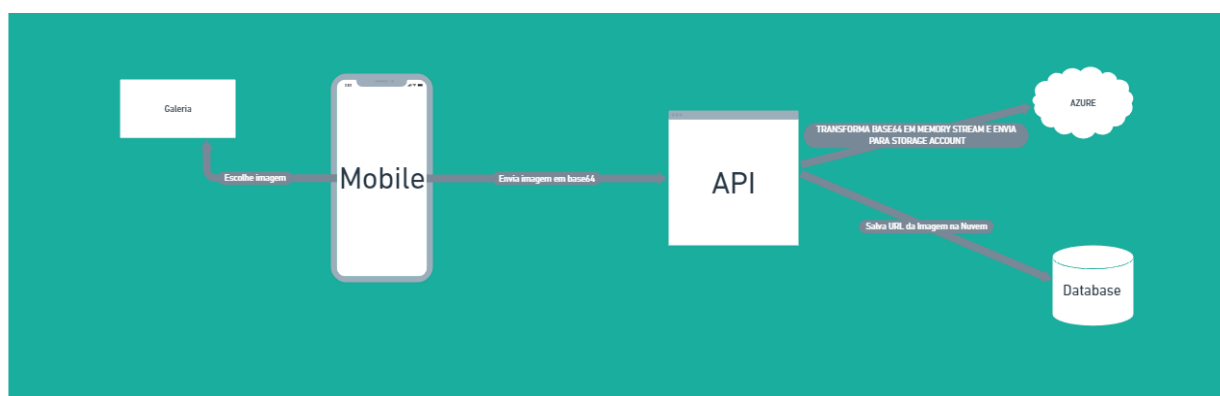


Figura 21: Fluxo de envio de imagem para o Azure.

Fonte: Autoria Própria, 2022

3.6.1.2 AZURE FUNCTIONS

O Azure Functions, segundo a Microsoft (2022), é um recurso sem servidor que permite escrever menos código, uma vez que deve ser focado apenas na regra

de negócio que precisa ser implementada, já que a nuvem vai fornecer a infraestrutura necessária para a execução da função.

Este recurso é bastante útil quando é necessário executar um código baseado em eventos, seja o evento por tempo, por requisição ou por processamento de filas. No caso do aplicativo que foi desenvolvido, é necessário que um algoritmo seja executado de hora em hora, buscando os Posts que o usuário cadastrou relatando a inacessibilidade de alguma via, e, caso o Post tenha sido criado há mais de 24 horas e tiver ao menos 70% mais aprovações sobre as reprovações, simbolizadas por “likes” e “deslikes”, então o post deve ser aprovado, caso contrário, o post será reprovado.

Neste caso, essa tarefa é baseada em um evento de tempo, com um intervalo de 1 hora. Identificado o evento, basta então criar um projeto do tipo Azure Functions e implementar a regra que deverá ser executada, informando também a periodicidade como parâmetro da função. A Figura 22 exhibe como foi feito o desenvolvimento do recurso:



```
[FunctionName("EvaluatePost")]
public static async void Run([TimerTrigger("0 * */1 * * *")] TimerInfo myTimer, ILogger log)
{
    log.LogInformation($"C# Timer trigger function executed at: {DateTime.Now}");
    string cs = Environment.GetEnvironmentVariable("MyConnectionString",
        EnvironmentVariableTarget.Process);

    var command1 = PublishPost();
    var command2 = RejectPost();
    string[] commands = { command1, command2 };

    using (SqlConnection conn = new SqlConnection(cs))
    {
        conn.Open();

        foreach (var command in commands)
        {
            using (SqlCommand cmd = new SqlCommand(command, conn))
            {
                // Execute the command and log the # rows affected.
                var rows = await cmd.ExecuteNonQueryAsync();
                log.LogInformation($"{rows} rows were updated");
            }
        }
    }
}
```

Figura 22: Azure Function para avaliar o Post.
Fonte: Autoria Própria, 2022

A Figura 22 ilustra o método Run, com uma anotação de *FunctionName* definida por *EvaluatePost*, que é o nome que será utilizado para identificar a função na Azure. A função recebe por parâmetro um objeto do tipo *TimerTrigger* que é o evento que será utilizado para disparar a função e que também necessita do parâmetro de configuração do tempo, nesse caso foi passado o texto (0 * */1 * * *) onde cada asterisco representa uma unidade de medida de tempo, começando no 0 temos os segundos, minutos, horas, dias, meses, ano.

Neste caso, foi definido um */1 onde a barra (/) significa intervalo, na posição das horas, registrando então que no intervalo de 1 hora, a função será utilizada. No corpo da função, foi definido um log que deixará registrado quando a função foi executada e logo em seguida é armazenada em uma variável o texto de conexão com o banco de dados, que será utilizado para abrir a conexão com o banco de

dados e executar o script de publicação do post como também o de rejeição do post.

Após o desenvolvimento da Função, é feita a publicação do projeto para o Azure, que então será responsável por disparar a Função a cada intervalo de 1 hora.

3.7 ARQUITETURA MOBILE

No desenvolvimento *mobile*, é adotada uma arquitetura minimalista com o objetivo de facilitar a compreensão do projeto e também de separar as responsabilidades, que são divididas por meio de pastas com nomenclaturas coerentes. A Figura 23 exibe a estrutura da aplicação mobile:

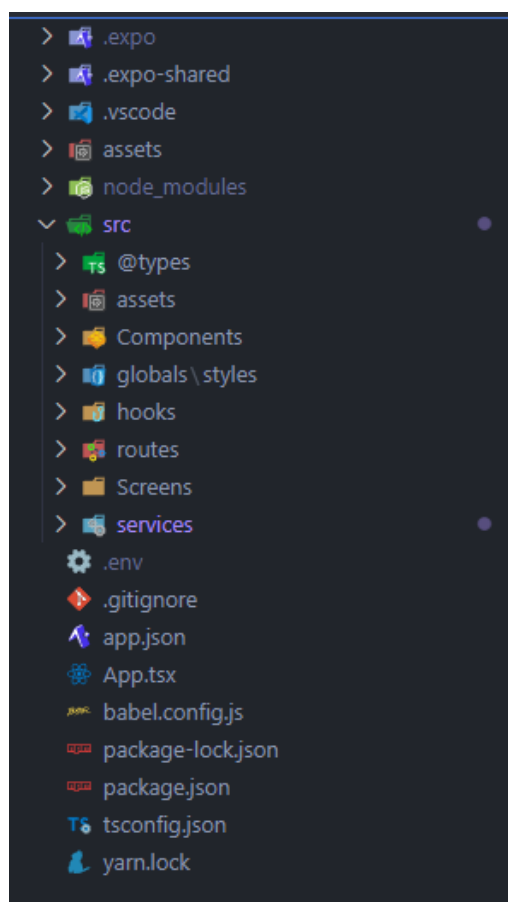


Figura 23: Arquitetura mobile.
Fonte: Autoria Própria, 2022

Como pode ser observado no canto superior da Figura 23, a pasta principal do sistema é a *src*, que contém toda a estrutura de código da aplicação. Dentro da pasta *src*, podemos observar as pastas:

- *@types*: responsável por definir os tipos de arquivos que a linguagem TypeScript não possui, como por exemplo arquivos de imagem png.
- *assets*: armazena os arquivos de mídia do aplicativo, podendo ser imagens e arquivos svg.
- *components*: gerencia os componentes da aplicação, estes podem ser compartilhados e reutilizados para evitar a duplicidade de código no sistema.
- *globals*: centraliza os arquivos que podem ser usados em qualquer parte do sistema, tais como estilização css e fontes.
- *hooks*: gerencia as funções de apoio da aplicação, tanto os que a aplicação disponibiliza quanto os que podemos implementar.
- *routes*: administra as rotas do sistema e também controla por meio da autenticação, quais as telas que o usuário pode acessar.
- *screens*: gerencia as telas do sistema, essas são: Sign In, Home, Forum, User Profile, Post Create e About Us.
- *services*: pasta que contém a configuração da ferramenta que será utilizada para fazer as requisições para o backend da aplicação, está contido também a URL de requisição para que não seja necessário informá-la em todas as páginas que precisarem fazer alguma requisição.

3.7.1 SINALIZAÇÃO DA ROTA NÃO ACESSÍVEL NO APLICATIVO

Quando a rota é cadastrada na tela de Novo *Post*, fazemos uso da API do google chamada de *Google Places*, que converte endereços em informações como latitude e longitude. Desta maneira, quando o usuário digita o ponto de partida e ponto de destino, o google converte esta informação para latitude e longitude e essas informações são enviadas para o back-end.

Caso a rota seja votada e publicada como não acessível, o aplicativo faz uso da biblioteca *PolyLines* que desenha linhas no mapa, e como parâmetros para as linhas, é enviado a latitude e longitude que foi cadastrada na rota.

A princípio, as *PolyLines* foram criadas para delimitar uma região no mapa, fazendo uma ligação entre os pontos e gerando uma forma geométrica. Para que esta biblioteca pudesse funcionar de maneira com que não ligasse os pontos, ao invés de passar por parâmetro apenas 1 objeto com várias latitudes e longitudes, é passado vários objetos com apenas 2 latitudes e 2 longitudes, sendo essas 1 para ponto de partida e a outra para ponto de destino, desta maneira é possível ter várias *PolyLines* sinalizando os trechos inacessíveis.

Esta biblioteca também conta com parâmetros de cor, largura da linha e a posição no eixo Z. No caso do aplicativo, é utilizado a cor vermelha, a largura é a 1 pixel mais grossa do que a padrão, para destacar, e por fim, a posição no eixo Z define se a *PolyLine* irá sobrepor a linha da rota ou não, no nosso caso ela fará a sobreposição para sinalizar que a rota passa por um trecho inacessível.

3.8 AUTENTICAÇÃO

Para a autenticação do sistema, optou-se pela estratégia do login social com a Google, com isso o sistema elimina o gerenciamento de usuário e desenvolvimento de serviços como: recuperação de senha e envio de SMS e pode focar no objetivo principal. Para que pudesse ser utilizado o login social, a autenticação foi dividida em duas partes que serão descritas abaixo:

3.8.1 LOGIN COM GOOGLE PELO APLICATIVO

Com o objetivo de autenticar o usuário pela conta da Google, foi criada uma conta na plataforma de desenvolvedores da Google e registrado o aplicativo. Ao registrar o aplicativo, recebe-se um código que identifica aquele aplicativo para a Google.

Uma vez recebido o código, deve-se criar as configurações de login e quais informações seu aplicativo poderá acessar do usuário, no caso deste aplicativo, foi escolhido informações de perfil e de e-mail e por meio do OAuth 2 que segundo a Força-Tarefa de Engenharia de Internet (2012) é um protocolo que permite que um terceiro aplicativo obtenha acesso limitado a um serviço HTTP, seja em nome de um proprietário de recurso (nesse caso a Google) orquestrando uma interação de aprovação entre o proprietário do recurso e o serviço HTTP, ou permitindo que o aplicativo de terceiros obtenha acesso em seu próprio nome, é garantido a autenticidade do usuário.

Para que esta estratégia funcione, primeiramente deve-se enviar uma requisição para a Google e nesta requisição é necessário informar qual o código do nosso aplicativo, assim como uma URL de redirecionamento que o Google enviará as informações e por fim o escopo de informações que desejamos acessar do usuário.

Caso nosso código esteja correto e o usuário se autentique de maneira correta, a Google enviará um token para a URL de redirecionamento informada na requisição, esse token será utilizado para buscar as informações do usuário, então uma nova requisição é necessária, mas desta vez apenas informando o token que a Google forneceu na primeira requisição, caso o token seja válido, será retornado as informações do usuário que foi pedido na primeira requisição, no nosso caso, as informações de *Profile* e Email.

Pode-se observar o fluxo de login social na Figura 24:

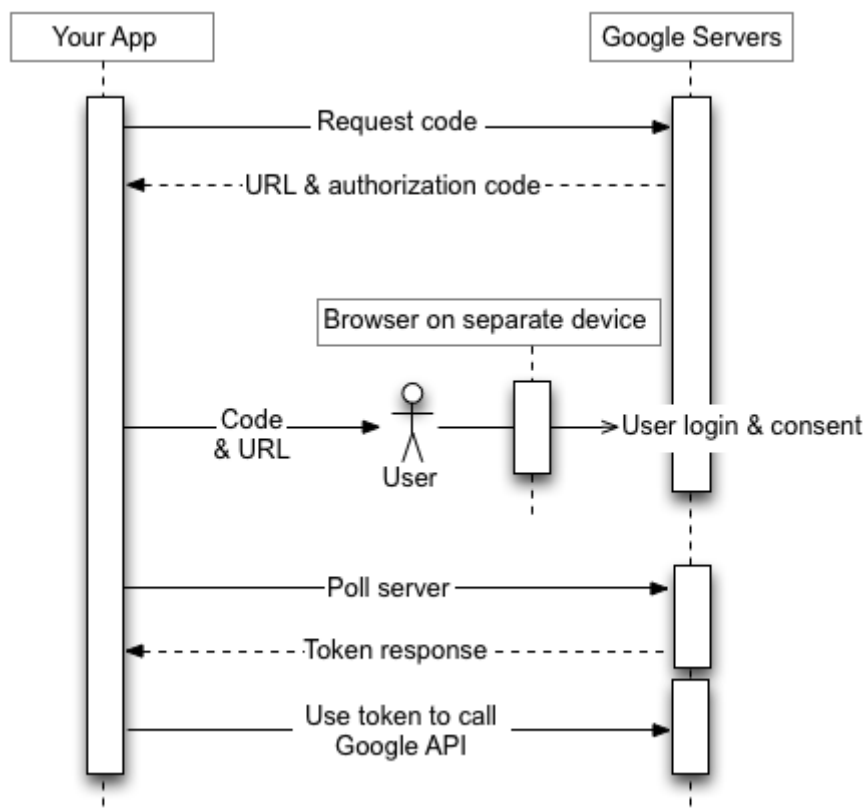


Figura 24: Login Social.
Fonte: DEVELOPERS, 2022

3.8.2 AUTENTICAÇÃO NO BACK-END

Nesta última etapa de autenticação, uma vez feito o login pelo Google e obtido os dados do usuário, é enviada a informação para o back-end da aplicação que faz o cadastro do usuário caso seja um email que não exista no banco de dados.

Após a etapa de cadastro, é seguido para a etapa de autenticação, é utilizado o email do usuário e o id da conta do mesmo no Google que só pode ser obtido a partir da autenticação com o mobile. Após verificado a autenticidade, é gerado um novo token do tipo JSON Web Token, popularmente conhecido como JWT, que será detalhado em seguida. Este token é retornado para o mobile e deve ser utilizado no cabeçalho de cada requisição, pois a única função que o usuário consegue fazer no sistema sem um token de autenticação é o Login.

Toda requisição feita após o login, vai esperar um token e vai validar se o token ainda está consistente, caso esteja consistente, o sistema permitirá que a ação seja executada.

Pode-se observar na Figura 25 o fluxo de autenticação após o login com o Google:

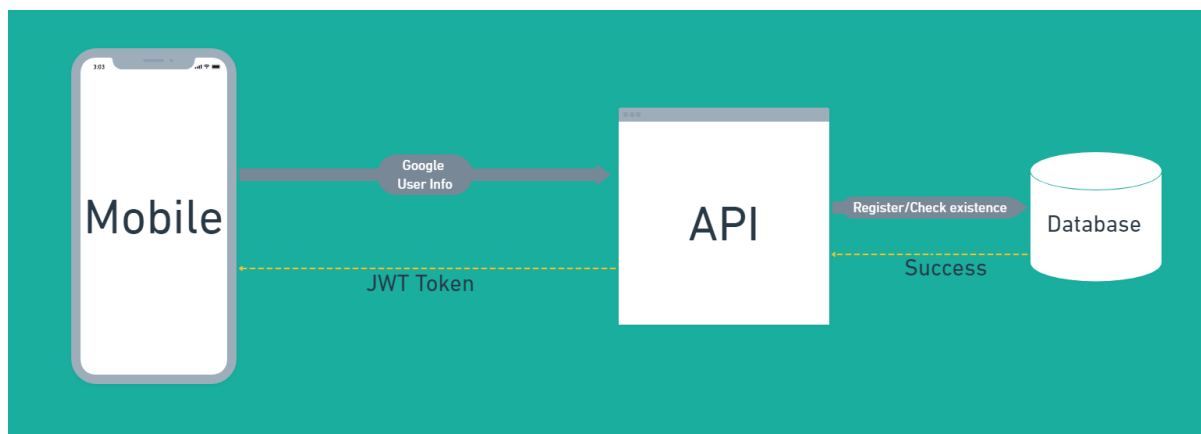


Figura 25: Fluxo de autenticação backend.
Fonte: Aatoria Própria, 2022

3.8.3 JSON WEB TOKEN

O JSON Web Token é um método de padrão aberto da indústria (2019) para realizar a autenticação entre as partes. A autenticação acontece por meio de um token assinado que garante a autenticidade da requisição. Ele é um objeto transformado em base64 e depois criptografado que contém informações em formato JSON com os dados que permitem verificar a autenticidade da requisição.

O JWT possui 3 componentes básicos, o Header que é o cabeçalho do token e contém duas informações, sendo ela o algoritmo que será utilizado para criptografar as informações, comumente sendo o HS256 (algoritmo de HASH utilizando uma codificação de 256 bits) e a segunda informação é padrão sendo o tipo do token, no caso, JWT.

O segundo componente de um JWT é o *payload* e este componente contém os dados do usuário que deseja se autenticar, como email e senha. O *payload* também pode conter informações adicionais, por exemplo o tempo de expiração do token.

Por fim, o terceiro componente do JWT é a *signature*, que basicamente depende do *header*, *payload* e de uma chave secreta, que é armazenada apenas na API e garante que só quem tem acesso a chave secreta, possa decodificar o Token. Para gerar a *signature* então devemos codificar o *header* e o *payload* em *base64* e usando o algoritmo que foi definido no *header* do JWT e mais a palavra chave, assinamos o token. O dado de saída então são 3 textos codificados em *base64* e divididos por ponto (.) que representam o token.

O fluxo básico de obtenção de um token pode ser visto na Figura 26:

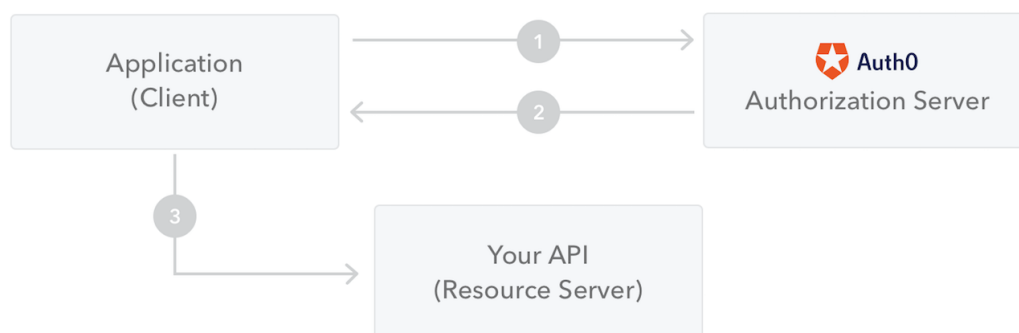


Figura 26: Fluxo de obtenção do Token.
Fonte: JWTIO, 2022

3.9 DOCUMENTAÇÃO DA API

Com o objetivo de tornar o uso da API mais fácil para quem vai consumir, faz-se necessário a criação de uma documentação. Com a documentação apropriada, é possível saber com antecedência quais rotas podem ser chamadas, qual parâmetro a rota espera e também qual o retorno que o consumidor pode esperar. É possível saber também quais são as entidades utilizadas no projeto, sendo essa uma informação opcional de quem está criando.

Para a documentação deste projeto, foi utilizado a ferramenta Swagger, que segundo Aristóteles (2019) é um framework para descrição, consumo e visualização de serviços RESTful que o grande objetivo é permitir que a documentação evolua paralelamente com o desenvolvimento, uma vez que a documentação pode ser gerada automaticamente de acordo com as anotações feitas em código.

O Swagger possui uma interface que permite que os desenvolvedores interajam com as APIs de maneira intuitiva, funcionando similarmente como um *HTTP Client*, utilizado comumente para fazer testes em APIs. Além de poder testar os endpoints das APIs, é possível visualizar todas as informações necessárias para realizar a requisição e o que esperar de retorno.

A Figura 27 exhibe, pela interface do Swagger, os detalhes da rota de criação de um Post, assim como os as propriedades esperadas dentro do corpo da requisição e também o tipo de resposta que será retornado, caso a requisição atenda com o requerido.

Na Figura 28 podemos observar a rota de consulta de um usuário específico, e pela interface do Swagger podemos observar quais são os parâmetros obrigatórios, no nosso caso seria o Id do usuário, para que a requisição seja realizada com sucesso e retorne o usuário esperado.

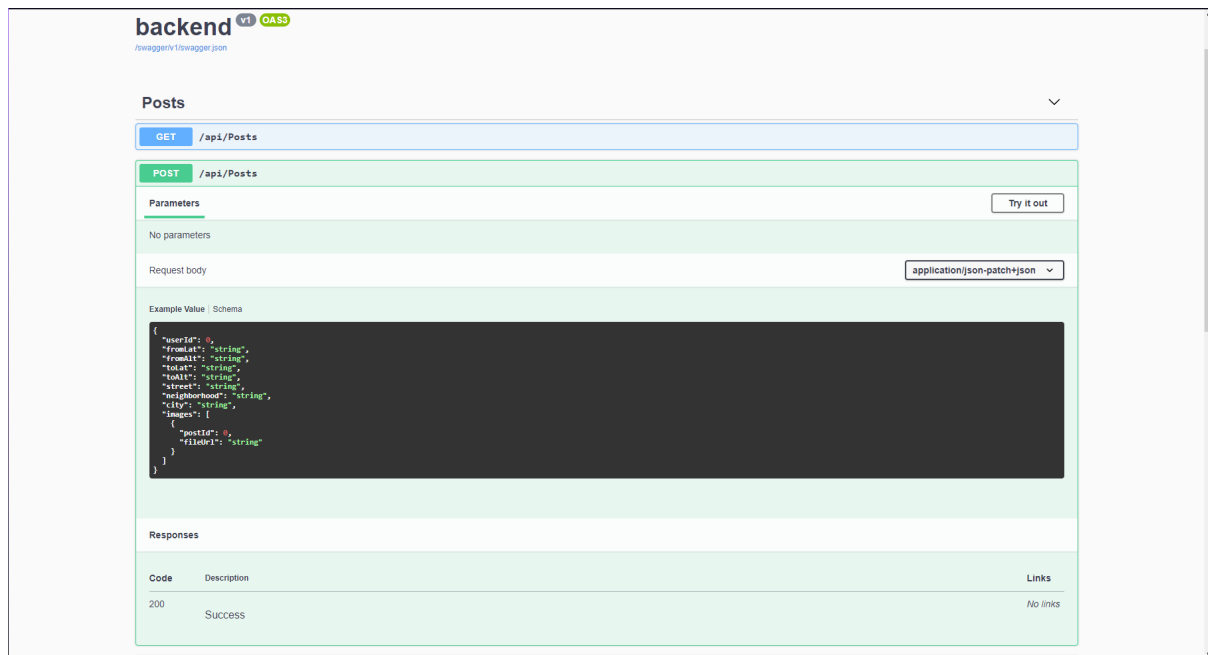


Figura 27: Swagger UI endpoint object required.
Fonte: Autoria Própria, 2022

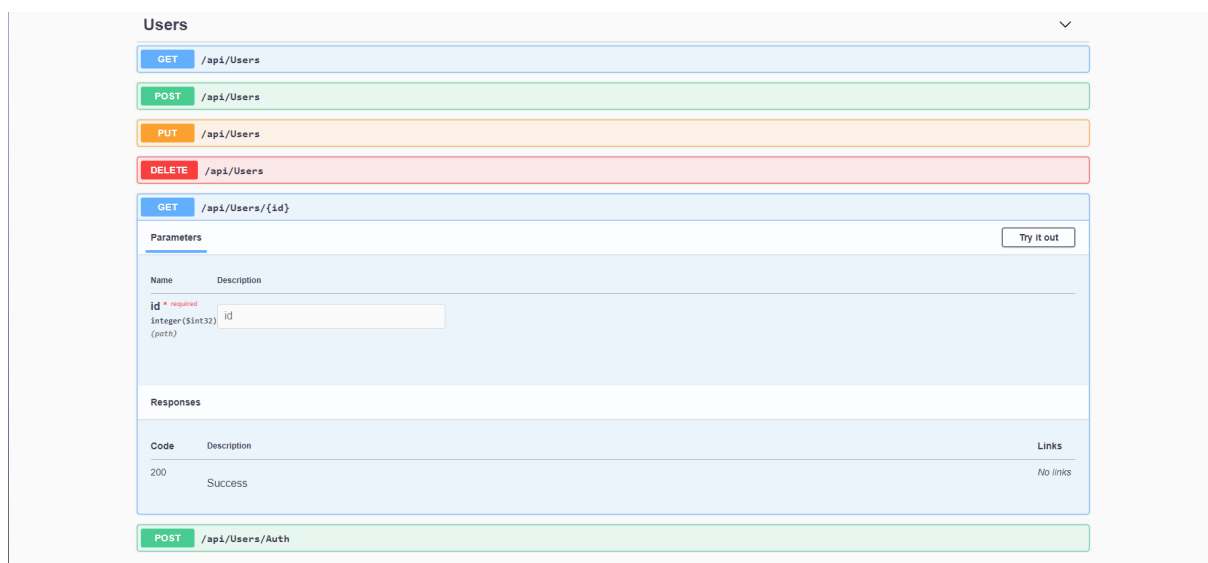


Figura 28: Swagger UI endpoint parameters required.
Fonte: Autoria Própria, 2022

4. CONCLUSÕES

4.1 LIMITAÇÕES

Com o fim do desenvolvimento do aplicativo, é observado que há algumas limitações referentes à direção da calçada, uma vez que uma rua pode ter duas calçadas em suas extremidades, porém pelo fato de estar sendo utilizado um mapa a nível de rua, não há como distinguir qual das duas calçadas está inacessível, mostra apenas que a rua está inacessível.

Caso alguma calçada que estava inacessível, passou por reestruturação e então se torne acessível, não será possível cadastrá-la como acessível, tendo que a prefeitura ou usuário fazer um reporte via redes sociais para que seja removido o registro de inacessível da calçada via banco de dados.

4.2 TRABALHOS FUTUROS

Até o presente momento, a google não possui uma API que faça o trabalho de “evitar uma rota específica”, sendo esta uma funcionalidade que acrescentaria muito no aplicativo, uma vez que atualmente o aplicativo traça a rota mesmo que esteja inacessível, utilizando como indicador, uma sobreposição de cores, a cor da rota é azul e a cor da rua inacessível é vermelha, caso a rota passe por um trajeto inacessível, o vermelho ficaria sobreposto ao azul, indicando que aquele trecho é inacessível.

O aplicativo não difere a inacessibilidade cadastrada no *post* por cores, se há alguma inacessibilidade, será colocada na cor vermelha. Neste cenário, como uma implementação futura, a possibilidade de diferenciar as inacessibilidades por cores, tais como: para a calçada sem rampa de acesso, utilizar a cor amarela. Para a calçada com obstáculos, manter a cor vermelha. Para a calçada sem nivelamento, utilizar a cor marrom.

Implementar a possibilidade de cadastrar também as calçadas que estão acessíveis.

Adicionar o gerenciamento de usuários em um banco de dados da própria aplicação, pois no aplicativo só é possível realizar a autenticação por meio do login social com a Google, e caso o usuário não possua um email da google, não conseguirá utilizar o sistema.

O aplicativo ficará disponível no repositório do Github², onde poderá ser feito as requisições para alteração e a contribuição coletiva.

4.3 CONSIDERAÇÕES FINAIS

Espera-se que o aplicativo desenvolvido seja de grande serventia para os usuários, que possa ajudar as pessoas com deficiência motora e ou mobilidade reduzida, que a população se mobilize e ajude no mapeamento da cidade.

É aguardado também que com a incorporação total do aplicativo na sociedade, que as prefeituras dos municípios se conscientizem e saibam quais os locais que estão trazendo déficit de locomoção, para que posteriormente tomem medidas corretivas e que dessa forma, tornem as cidades cada vez mais acessíveis, servindo como ponto de inspiração para as demais cidades.

Por fim, espera-se que o aplicativo traga segurança para os deficientes e seja um ponto de partida para um mundo mais acessível.

² <https://github.com/maumauagain/RoutesApp>

REFERÊNCIAS

DEVMEDIA, DTO. Diferença entre os patterns PO, POJO, BO, DTO e VO. Disponível em:

<<https://www.devmedia.com.br/diferenca-entre-os-patterns-po-pojo-bo-dto-e-vo/28162>>

Acesso em: 7 de setembro de 2022.

DEVMEDIA, JWT. O que é JSON Web Token?. Disponível em:

<<https://www.devmedia.com.br/como-o-jwt-funciona/40265>> Acesso em: 7 de setembro de 2022.

GALLARDO, Claudia Pradas. Significado da cor vermelha na psicologia, 2019. Disponível

em: <<https://br.psicologia-online.com/significado-da-cor-vermelha-na-psicologia-197.html>>

Acesso em: 23 de outubro de 2022.

GOOGLE, API do Google Maps. Disponível em:

<<https://cloud.google.com/maps-platform/maps/>> Acessado em: 17 de outubro de 2019.

GOOGLE, Google Maps. **Inscrição para empréstimo de câmera do Street View.**

Disponível em: <<https://www.google.com/intl/pt-BR/streetview/loan/>> Acesso em: 4 de junho de 2019.

GOOGLE, Waze. Como o Waze funciona?. Disponível em:

<https://support.google.com/waze/answer/6078702?hl=pt-BR&ref_topic=9022747> Acesso em: 17 de outubro de 2019.

IBGE, IBGE. PESSOAS COM DEFICIÊNCIA. Disponível em:

<<https://educa.ibge.gov.br/jovens/conheca-o-brasil/populacao/20551-pessoas-com-deficiencia.html>> Acesso em: 23 de outubro de 2022.

JWT, JWT. Introduction to JSON Web Tokens. Disponível em: <<https://jwt.io/introduction>>

Acesso em: 7 de setembro de 2022.

LI, He.; ZHIJIAN, Lai. The study and implementation of mobile GPS navigation system based on Google Maps, IEEE 2012. In 2010 International Conference on Computer and Information Application. Disponível em: <<https://ieeexplore.ieee.org/document/6141544>>
Acesso em: 03 de junho de 2019.

LOPES, Aristóteles. Porque usar Swagger na sua API?. Disponível em:
<<https://medium.com/@arikardnoir/porque-usar-swagger-na-sua-api-e80c4ed15190>>
Acesso em: 11 de setembro de 2022.

MATHARU, Gurpreet Singh et al. Empirical study of agile software development methodologies: A comparative analysis. ACM SIGSOFT Software Engineering Notes, v. 40, n. 1, p. 1-6, 2015.

MICROSOFT, Azure Functions. Introduction to Azure Functions. Disponível em:
<<https://learn.microsoft.com/en-us/azure/azure-functions/functions-overview>> Acesso em: 10 de outubro de 2022.

MICROSOFT, Azure Storage Account. Visão geral da conta de armazenamento. Disponível em:
<<https://learn.microsoft.com/pt-br/azure/storage/common/storage-account-overview>>. Acesso em: 10 de outubro de 2022.

MOBILIZE, Mobilidade urbana sustentável. **Levantamento "Calçadas do Brasil"**. Disponível em: <<http://www.mobilize.org.br/campanhas/calçadas-do-brasil/levantamento>>. Acesso em maio de 2019.

NUNES LEAL, Luciana.; THOMÉ, Clarissa. **Brasil tem 45,6 milhões de deficientes**, 2012. Disponível em:
<<https://www.estadao.com.br/noticias/geral,brasil-tem-45-6-milhoes-de-deficientes.893424>>. Acesso em: 03 de junho de 2019.

OTT do Brasil. Onde tem Tiroteio. Disponível em: <<https://www.ondetemtiroteio.com.br/>>

Acesso em: 1 de outubro de 2019.

PREFEITURA SÃO PAULO, PREFEITURA SÃO PAULO. Símbolos de Acessibilidade.

Disponível em:

<https://www.prefeitura.sp.gov.br/cidade/secretarias/pessoa_com_deficiencia/a_imprensa/index.php> Acesso em: 23 de outubro de 2022.

PRESIDÊNCIA DA REPÚBLICA, CASA CIVIL, SUBCHEFIA PARA ASSUNTOS

JURÍDICOS. **Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência)**. Disponível em:

<http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm>. Acesso em maio de 2019.

RFC, Força-Tarefa de Engenharia de Internet. The OAuth 2.0 Authorization Framework.

Disponível em: <<https://www.rfc-editor.org/rfc/rfc6749>> Acesso em: 7 de setembro de 2022.

Souza, Ludmilla. Agência Brasil. **Pessoas com deficiência física criticam falta de acessibilidade em SP**. Disponível em:

<<http://agenciabrasil.ebc.com.br/geral/noticia/2019-09/pessoas-com-deficiencia-fisica-critica-m-falta-de-acessibilidade-em-sp>> Acesso em: 18 de outubro de 2019.

SRIVASTAVA, Apoorva.; BHARDWAJ, Sukriti.; SARASWAT, Shipra. **SCRUM model for agile methodology**, IEEE 2017. In International Conference on Computing, Communication and Automation (ICCCA). Disponível em: <<https://ieeexplore.ieee.org/document/8229928>>

Acesso em: 03 de junho de 2019.

TERLECKA, Katarzyna. Combining Kanban and Scrum -- Lessons from a Team of Sysadmins, IEEE 2012. In 2012 Agile Conference. Disponível em:

<<https://ieeexplore.ieee.org/document/6298098>> Acesso em: 3 de junho de 2019.