

2019

Relatório de Engenharia do Sistema Eletrônico de Gestão do Transporte Escolar (SETE)

*Relatório de Engenharia do
Sistema Eletrônico de Gestão do
Transporte Escolar (SETE)*



Universidade Federal de Goiás - UFG

Reitor

Prof. Dr. Edward Madureira Brasil

Faculdade de Ciências e Tecnologia – FCT

Diretor

Prof. Dr. Júlio Cesar Vallandro Soares

Coordenador do Projeto

Prof. Dr. Willer Luciano Carvalho

Equipe de Técnica

Prof. Dr. Marcos Paulino Roriz Junior
Prof. Dr. Paulo Henrique Cirino Araújo
Profa. M.Sc. Poliana de Sousa Leite
Prof. Dr. Ronny Marcelo A. Medrano
Profa. Dra. Waleska de Fátima Monteiro
Profa. Dra. Yaeko Yamashita
Beatriz Oliveira Leles de Faria
Bruna Borges Azevedo
Bruno Fernandes Freire
Gabriela Teixeira de Sousa Rosa
João Paulo Batista de Sousa
José Umberto Moreira
Jordana Alves Felipe
Letícia Gonçalves Silva
Lorena Furtado Borges
Maurício Barbosa da Cruz
Paulo Júnio Moura Rosa
Sara Santos de Almeida

Fundo Nacional de Desenvolvimento da Educação-FNDE

Presidente

Rodrigo Sergio Dias

Diretora da Diretoria de Ações Educacionais – DIRAE

Karine Silva dos Santos

Coordenadora-Geral de Apoio à Manutenção Escolar– CGAME

João Antônio Lopes de Oliveira

Coordenador de Apoio ao Transporte do Escolar – COATE
Gabriel Marques Andreozzi

Coordenador de Apoio ao Caminho da Escola – COACE
Djailson Dantas de Medeiros

Sumário

Sumário	4
Lista de Figuras	6
Lista de Tabelas.....	7
Apresentação	8
1. Introdução	10
2. Arquitetura do Sistema.....	17
2.1. Requisitos.....	17
2.2. Arquitetura do Software	19
2.3. Framework	20
2.4. Tecnologias de Informação Geográfica	22
3. Gestão de Dados.....	26
3.1. Modelo Conceitual	26
3.2. Modelo Lógico	28
3.2.1. Escolas.....	30
3.2.2. Estudantes.....	31
3.2.3. Motoristas.....	33
3.2.4. Veículos.....	35
3.2.5. Rotas.....	37
3.2.6. Garagem	38
3.2.7. Fornecedor.....	39
3.3. Tecnologias de Armazenamento	40
4. Gestão de Rotas.....	43
4.1. Cadastro de Rotas.....	43
4.2. Importar Rotas.....	44
4.3. Desenho de Rotas	45
4.4. Sugestão de Rotas (Roteirização).....	47
4.4.1. Aglomeração (<i>k-Means</i>).....	49
4.4.2. Clarke e Wright	50
4.4.3. Busca Local (2-opt).....	54
4.4.4. Resultado da Roteirização.....	55
5. Relatórios	57
5.1. Estudantes	58

5.2.	Escolas.....	59
5.3.	Motoristas.....	59
5.4.	Veículos.....	60
5.5.	Rotas.....	60
5.6.	Garagem	61
5.7.	Fornecedor.....	61
6.	Considerações Finais.....	62
	Referências Bibliográficas	63

Lista de Figuras

Figura 1.1: Tela principal do modo desktop do software SETE	12
Figura 1.2: Telas do modo móvel do software SETE	13
Figura 1.3: Módulos do software SETE	14
Figura 2.1: Arquitetura do <i>software</i> SETE	18
Figura 2.2: Estrutura do <i>framework</i> Electron	21
Figura 2.3: Exemplo de utilização da tecnologia OpenStreetMaps para cadastro da localização de usuários	22
Figura 2.4: Exemplo de utilização do <i>software</i> livre de edição de malha S2EM	23
Figura 3.1: Modelo conceitual do banco de dados do sistema	26
Figura 3.2: Modelo lógico (e simplificado) do banco de dados do sistema	28
Figura 4.1: Tela de cadastro de dados de rota	43
Figura 4.2: Tela de importação de rota a partir de dados GPX	44
Figura 4.3: Tela de desenho e edição de rotas	45
Figura 4.4: Tela de parametrização da roteirização	46
Figura 4.5: Etapas do algoritmo de roteirização	47
Figura 4.6: Exemplo de execução do algoritmo de aglomeração <i>k-Means</i>	48
Figura 4.7: Algoritmo <i>k-Means</i>	49
Figura 4.8: Cenário exemplo do algoritmo Clarke e Wright	50
Figura 4.9: Três cenários de combinação de rota do algoritmo Clarke e Wright	51
Figura 4.10: Exemplo de cenário onde a inversão de rota é vantajosa	52
Figura 4.11: Exemplo de uso do algoritmo 2-opt	53
Figura 4.12: Tela de resultado de sugestão de rotas	55
Figura 5.1: Tela de relatórios de alunos	56

Lista de Tabelas

Tabela 3.1 Esquema de dados da entidade Escola.	30
Tabela 3.2 Esquema de dados da entidade Estudante.	32
Tabela 3.3 Esquema de dados da entidade Motorista.	34
Tabela 3.4 Esquema de dados da entidade Veículo.	35
Tabela 3.5 Esquema de dados da entidade Rota.	37
Tabela 3.6 Esquema de dados da entidade Garagem.	38
Tabela 3.7 Esquema de dados da entidade Fornecedor.	39
Tabela 3.8 Esquema de dados da entidade OrdemDeServico.	40

Apresentação

Este relatório tem como objetivo a apresentação do projeto de engenharia de software do Sistema Eletrônico de Gestão do Transporte Escolar (SETE). O *software* foi projetado para auxiliar na gestão do transporte escolar nos municípios brasileiros.

Especificamente, apesar dos avanços das Tecnologias de Informação e Comunicação (TICs) nos municípios brasileiros, nota-se um aumento tímido na utilização de sistemas de informação de *e-governança* direcionados a gestão do transporte escolar (FERREIRA, 2016; PRZEYBILOVICZ; CUNHA; MEIRELLES, 2018). Entre os diversos motivos para a baixa adoção, encontra-se o custo de licenciamento e/ou desenvolvimento, além das dificuldades em implantar e utilizar os sistemas existentes efetivamente (CUNHA; QUANDT, 2014; JAMBEIRO; SOBREIRA; RABELO, 2009).

A principal dificuldade decorre de os sistemas existentes serem voltados a um aspecto único da gestão do transporte escolar, tais como a roteirização e/ou gestão de frotas, sendo poucos os sistemas que possuem uma visão unificada dos diversos atores do transporte escolar (FÁTIMA, 2015; FERREIRA, 2016; PORTO *et al.*, 2015). Isto dificulta a utilização dos sistemas, não somente por ter de implementar diferentes tecnologias, mas por ter de replicar os dados em cada um deles.

Outra dificuldade decorre dos sistemas serem projetados para municípios com alto nível de maturidade e recursos. Por exemplo, alguns sistemas requerem o georeferenciamento, *i.e.*, a localização GPS de todos os estudantes transportados para funcionar (MOREIRA; JÁCOMO; GRIEBELER, 2016; PORTO *et al.*, 2015). A obtenção desta informação não é algo simples para muitos municípios brasileiros. Como consequência, estes fatores podem dificultar a efetiva implantação dos sistemas nos mesmos.

Outra complexidade origina-se das licenças de *software*. Apesar de muitos sistemas serem disponibilizados como *software livre* gratuito, eles comumente dependem de sistemas proprietários para funcionar de forma adequada, como ArcGIS (ESRI, 2019) e Google Maps Cloud (GOOGLE, 2019a), cujo valores de custo podem ser exorbitantes para a realidade de diversos municípios brasileiros.

Com o intuito de abordar as limitações expressas, o Fundo Nacional de Desenvolvimento da Educação (FNDE) firmou parceria com o Centro Colaborador de Apoio ao Transporte Escolar da Universidade Federal de Goiás (CECATE – UFG) para o projeto e construção de um *software* voltado a gestão das várias facetas do transporte escolar no município.

Diante disso, foi firmado um Termo de Execução Descentralizada, de número TED Simec 6372, processo número 23400.000554/2017-56, entre o FNDE e a Universidade Federal de Goiás - UFG, por meio do curso de graduação em Engenharia de Transportes, da Faculdade de Ciências e Tecnologia. O projeto encontra-se cadastrado na UFG com número PI01834-2017 e no SEI/UFG como processo número 23070.102449/2017-30, e é denominado “Ações para o Desenvolvimento do Transporte Escolar no Brasil”.

Desta forma, o CECATE – UFG apresenta o Sistema Eletrônico de Gestão do Transporte Escolar (SETE), um *software* de *e-governança* de gestão do transporte escolar para municípios com níveis de gestão diferentes. O SETE unifica diversos aspectos realizados isoladamente por outros sistemas, como a gestão de frota, a gestão de estudantes e a roteirização. Note que o *software* SETE não tem o intuito de substituir os sistemas existentes, mas de complementá-los nos municípios que utilizam sistemas voltados a uma única faceta da gestão e de prover uma solução que se adeque a realidade dos municípios não atendidos pelos mesmos.

1. Introdução

A Constituição Federal Brasileira (BRASIL, 1988) assegura o direito ao transporte escolar público como meio de prover acesso universal à educação básica a todos os estudantes. Tal direito deve ser provido independentemente da localização do aluno, seja ele residente de uma área rural, urbana, ribeirinha, *etc.* Este requisito eleva a complexidade de realizar o transporte escolar, uma vez que as áreas rurais e remotas são carentes em termos de infraestrutura (FNDE/CEFTRU, 2007).

Como complicador, os municípios brasileiros são ditos predominantemente rurais. De fato, segundo estudo do IBGE (2017), 60.2% dos municípios se encaixam nesta categoria, destes 54,6% são classificados como rurais adjacentes e 5,8% como rurais remotos. A primeira categoria refere-se àqueles que possuem baixo grau de urbanização, mas que são adjacentes a municípios urbanos. Já a segunda categoria reflete os municípios longe de centros urbanos e com um índice igual ou inferior de ocupação aos rurais adjacentes.

A natureza rural dos municípios e, conseqüentemente, a falta de infraestrutura dificulta não somente a operação do transporte escolar, mas também o planejamento, uma vez que o mesmo deve atender estudantes com realidades distintas. Por exemplo, deve-se planejar rotas com horários distintos para atender tanto estudantes que moram nas extremidades dos municípios quanto aqueles que moram próximos as escolas (CARVALHO; LEITE; NASCIMENTO, 2015).

Em contraposto as dificuldades econômicas em investir e melhorar a infraestrutura viária, os municípios brasileiros têm adotado Tecnologias de Informação e Comunicação (TICs) com intuito de mitigar os problemas e otimizar a gestão de vários serviços prestados à população, incluindo o transporte escolar (CUNHA; QUANDT, 2014; FÁTIMA, 2015). O acesso à Internet é uma das condições básicas para o emprego das TICs. Felizmente, a pesquisa do Comitê Gestor da Internet no Brasil (CGI.BR, 2018) relata que a Internet se encontra presente de maneira universal nas prefeituras brasileiras.

Considerando a presença de uma infraestrutura de Internet, os municípios podem pensar em empregar TICs de governança eletrônica (*e-governança*) para melhorar os serviços prestados (JAMBEIRO; SOBREIRA; RABELO, 2009; PRZEYBILOVICZ; CUNHA;

MEIRELLES, 2018). Por exemplo, pode-se implantar sistemas de informação para monitorar o deslocamento dos veículos, ou sistemas para mapear a cobertura atual das rotas realizadas (FÁTIMA, 2015; MOREIRA; JÁCOMO; GRIEBELER, 2016).

Apesar da universalização da Internet nos municípios e avanços das TICs, observa-se um aumento tímido na utilização de sistemas de informação de *e-governança* nos mesmos, em especial de sistemas direcionados a gestão do transporte escolar (FERREIRA, 2016; PRZEYBILOVICZ; CUNHA; MEIRELLES, 2018). Pode-se elencar diversos motivos para a baixa adoção, desde dos custos de licença e/ou desenvolvimento de *software*, até a dificuldade de implantação e utilização do sistema pelos gestores do município (CUNHA; QUANDT, 2014; JAMBEIRO; SOBREIRA; RABELO, 2009).

Com intuito de abordar alguns desses problemas, a comunidade acadêmica e a indústria desenvolveram diversos sistemas de informação voltados a gestão (*e-governança*) do transporte escolar (FÁTIMA, 2015; FERREIRA, 2016; PORTO *et al.*, 2015). Entretanto, o foco de tais *softwares* é tipicamente restrito a um único aspecto da gestão, como a roteirização ou controle de frota, sendo que poucos sistemas se apresentam de forma unificada. Isto dificulta o emprego efetivo de tais *softwares* nos municípios, não somente pela complexidade em ter de aprender diversos sistemas, mas pela necessidade de ter de replicar dados existentes entre eles.

Além disso, os sistemas existentes são direcionados a municípios que possuem uma maturidade em termos de gestão, uma vez que requerem múltiplos dados como entrada. Por exemplo, como requisito para funcionar, alguns sistemas requerem como entrada o georeferenciamento de todos os estudantes, *i.e.*, a localização da residência dos mesmos em pares de latitude e longitude (MOREIRA; JÁCOMO; GRIEBELER, 2016; PORTO *et al.*, 2015). Além da complexidade de obtenção desses dados, estes requisitos funcionam como um entrave ao funcionamento dos sistemas nos municípios brasileiros, uma vez que uma parte significativa dos mesmos não possuem tais dados e, conseqüentemente, não poderão utilizar o *software*.

Também, apesar dos sistemas serem comumente disponibilizados como *software* livre, os mesmos tipicamente dependem de *softwares* proprietários (MOREIRA; JÁCOMO;

GRIEBELER, 2016). Dentre tais dependências, percebe-se comumente a utilização dos *softwares* ArcGIS (ESRI, 2019) e Google Maps Cloud (GOOGLE, 2019a), cujo valores de custo podem ser exorbitantes para a realidade de diversos municípios brasileiros. Outro fator complicador decorre da complexidade de licenciar tais sistemas dentro do arcabouço de licitação, uma vez que o modelo de negócio desses *softwares* é feito sob demanda. Por exemplo, paga-se o serviço do Google Maps Cloud com base no número de usuários que utilizam o sistema. Por fim, tais sistemas dependentes também podem requerer uma taxa mensal de mensalidade, uma vez que poucos fornecem uma licença de uso perpétua.

Com intuito de abordar os problemas apontados, o CECATE UFG desenvolveu o Sistema Eletrônico de Gestão do Transporte Escolar (SETE), um *software* de *e-governança* de gestão do transporte escolar para municípios com diferentes realidades. O SETE unifica diversos aspectos realizados isoladamente por outros sistemas, como a gestão de frota, a gestão de estudantes e a roteirização. Note que o *software* SETE não tem o intuito de substituir os sistemas mencionados, mas de complementá-los nos municípios que os utilizam com outras funções e de preencher uma lacuna ao prover uma solução unificada que se adeque a realidade dos municípios não atendidos pelos mesmos.

As ferramentas do SETE são fornecidas em diferentes modos para que possam ser usadas por municípios com realidades distintas. Por exemplo, um município que possui menos informações pode desejar apenas cadastrar informações a respeito das rotas existentes, como quilometragem e alunos atendidos. Já um município com uma gestão mais madura pode desenhar o traçado das rotas em mapas para melhor compreender os deslocamentos. Por fim, um município com maiores recursos pode buscar otimizar as rotas existentes, uma tarefa que necessita primeiramente passar pelo georeferenciamento dos estudantes.

Apesar da universalidade da Internet, sabe-se da precariedade da conexão nos municípios brasileiros (AMÉRICO, 2016; CGI.BR, 2018). Por essa razão, é desejável que o *software* SETE possa lidar com uma conexão intermitente com a Internet. Desta forma, o sistema é disponibilizado em dois modos: *desktop* e móvel. O primeiro modo (*desktop*) é a versão principal do sistema e será nela que o usuário realizará as ações de gestão do transporte escolar. Esta versão poderá ser utilizada tanto de maneira off-line (sem Internet), quanto

on-line (com Internet). Quando o usuário estiver conectado é feita uma sincronização dos dados e ações realizados com um banco de dados em nuvem.

A Figura 1.1 ilustra a tela principal do modo *desktop* do *software* SETE no sistema operacional Microsoft Windows 10. Observa-se que apesar do sistema ser executado localmente, a interface gráfica do mesmo se assemelha àquelas usadas por sítios da Internet. A semelhança é causada pela utilização do arcabouço (*framework*) Electron, que possibilita usar linguagens de programação e bibliotecas Web para construção de sistemas *desktop* (JENSEN, 2017). Ao adotar esse *framework*, pode-se facilitar a reutilização do código desenvolvido para uma versão future e Web do sistema, uma vez que o código desenvolvido pode ser executado diretamente nos navegadores dos usuários.

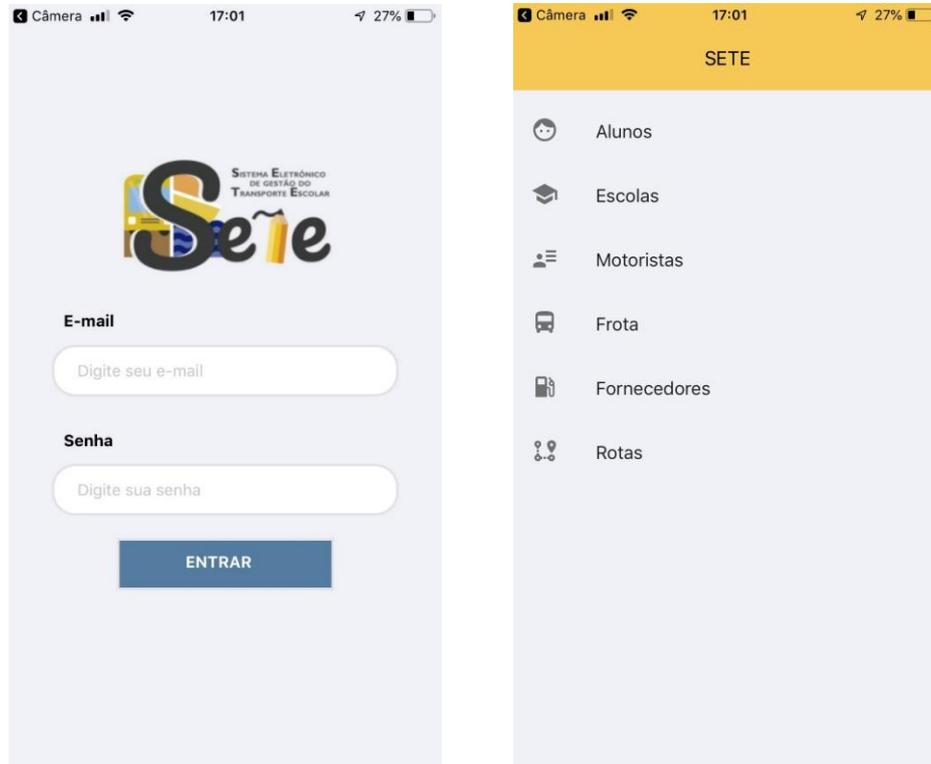
Figura 1.1: Tela principal do modo *desktop* do *software* SETE



Fonte: Elaborada pelos autores.

Já a versão móvel do *software* SETE, visa disponibilizar as informações do transporte escolar para os gestores via *smartphone*. Através do aplicativo móvel é possível visualizar as ações feitas e informações previamente cadastradas no modo *desktop*, tais como rotas traçadas, alunos atendidos e quilometragem dos veículos. O aplicativo móvel serve como um espelho que reflete os dados, salvos em nuvem, previamente cadastrados na versão *desktop*. A Figura 1.2 ilustra duas telas do aplicativo móvel na plataforma iOS. A primeira apresenta a tela de entrada, para que o usuário indique as suas credenciais, enquanto que

Figura 1.2: Telas do modo móvel do *software* SETE.



Fonte: Elaborada pelos autores.

a segunda apresenta uma tela com uma lista de atores do transporte escolar. O gestor pode clicar em cada um deles para obter um detalhamento dos mesmos.

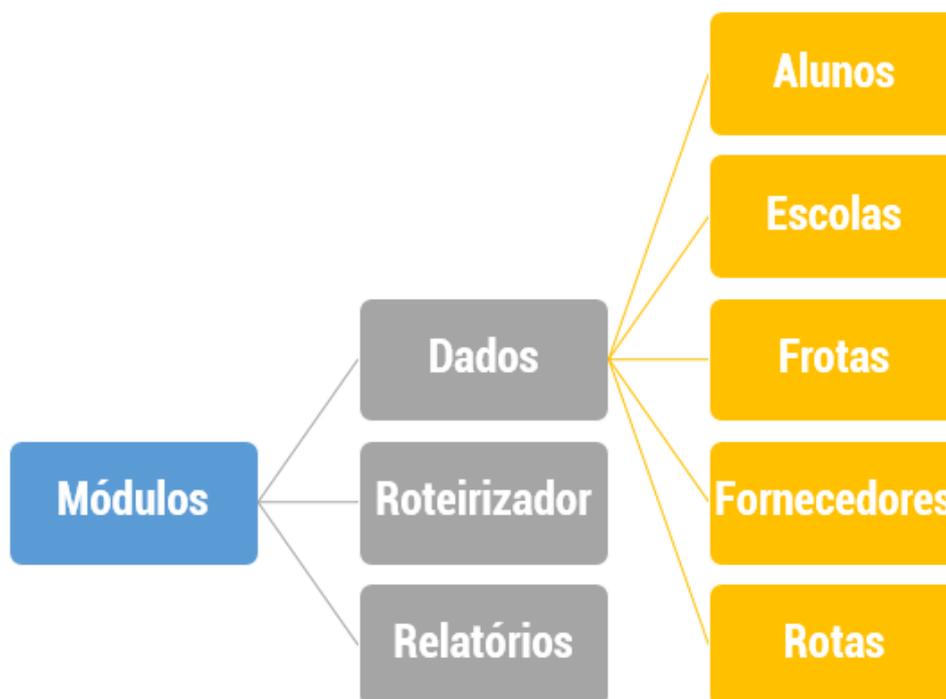
O SETE é distribuído sob a licença de *software* livre MIT (OSI, 2019a), que possibilita não somente a redistribuição gratuita, mas o compartilhamento e modificação do código fonte do mesmo por parte dos municípios. Por exemplo, o município que possuir uma equipe de tecnologia de informação pode realizar adaptações no programa e distribuí-las para outras cidades. Além disso, tomou-se o cuidado no desenvolvimento do sistema SETE para que o mesmo não possua nenhuma dependência com *softwares* proprietários, isto é, os municípios não precisam se preocupar em ter de lidar com licenciamento de outros sistemas de *software*.

Vale salientar que, com intuito de atender municípios que utilizam diferentes sistemas operacionais, o *software* SETE será disponibilizado em múltiplas plataformas. A versão

desktop está disponível para os sistemas operacionais Microsoft Windows, Mac OS X, e GNU/Linux, já o modo móvel atende as plataformas iOS e Android.

A arquitetura do SETE é dividida em módulos, conforme ilustrado pela Figura 1.3. O módulo de dados concentra as ferramentas do sistema para cadastrar, remover, atualizar, buscar e gerir os diferentes atores do transporte escolar. Já o roteirizador, utiliza os dados cadastrados, em conjunto com algoritmos de otimização, para sugerir automaticamente as rotas para os municípios. Observe que o georeferenciamento dos atores é pré-requisito para poder usar tal módulo. Por fim, o módulo de relatório visa sintetizar e extrair informações a partir dos dados inseridos.

Figura 1.3: Módulos do *software* SETE



Fonte: Elaborada pelos autores.

Para compreensão do funcionamento de cada módulo é necessário conhecer os requisitos e arquitetura do *software* utilizado. Desta forma, a próxima seção apresenta uma descrição sistêmica das funcionalidades do SETE, detalhando o papel e as interações dos módulos, tanto na versão *desktop*, quanto na versão móvel do sistema. Além disso, é feito um destaque das tecnologias utilizadas para realizar a programação das funcionalidades dos

módulos descritos. Especificamente, é feito um detalhamento das escolhas de projetos feitas para a implementação de cada módulo do sistema.

Logo em seguida, é feito um relato do modelo conceitual e lógico do banco de dados do projeto, com intuito de possibilitar o entendimento de quais serão os dados necessários para o funcionamento do sistema. A compreensão do esquema do banco de dados adota também possibilita compreender como os mesmos poderão ser reutilizados por outros sistemas.

Em seguida é feito uma descrição dos diversos mecanismos utilizados para gerir as rotas do transporte escolar. Destaca-se aqui a possibilidade do município de utilizar gerir-las em diferentes níveis. Por exemplo, o nível mais simples restringe-se ao cadastro de dados básicos das rotas, como quilometragem e estudantes atendidos. Níveis intermediários possibilitam a importação de rotas de dados GPS e o desenho das mesmas. Já o nível mais avançado possibilita utilizar algoritmos de roteirização para sugerir automaticamente as rotas para o gestor.

Os relatórios estão entre os principais documentos utilizados por gestores para a tomada de decisão. Nesse sentido, o *software* de *e-governança* SETE dedicou-se a produção de indicadores e gráficos estatísticos para todos os atores do transporte escolar. Destaca-se a capacidade do sistema de exportar as informações para planilhas eletrônicas e documentos no formato PDF.

Por fim, é feito as considerações finais e projetos de trabalhos futuros acerca do *software* SETE

2. Arquitetura do Sistema

As restrições dos *softwares* existentes voltados a gestão do transporte escolar, tais como a quantidade necessário de dados de entrada e o foco direcionado a uma faceta da gestão guiam o projeto arquitetural do *software* de *e-governança* SETE. Esta seção visa relatar como tais limitações foram abordadas na concepção do sistema SETE. Também se faz uma descrição do funcionamento e interação entre os módulos da arquitetura do *software*.

2.1. Requisitos

Os requisitos guiam o projeto arquitetural de um sistema de *software* (SOMMERVILLE; SAWYER, 1997). Eles descrevem o funcionamento esperado de um sistema. Podem ser divididos em requisitos funcionais e não-funcionais, sendo os primeiros vinculados ao comportamento das funções providas (*e.g.*, permitir cadastrar a localização de estudante e buscar as escolas atendidas), enquanto os segundos estão associados a critérios de uso do sistema, tipicamente descritos em termos de desempenho e usabilidade (*e.g.*, permitir a utilização em diferentes sistemas operacionais).

Desta forma, utilizou-se as limitações relatadas dos sistemas pré-existentes como guia para a descrição dos requisitos a serem abordados pelo *software* de *e-governança* SETE. Considerando a necessidade de prover um sistema que atenda municípios com realidades distintas, o SETE identificou os seguintes requisitos funcionais:

- **Gerir os dados dos atores do transporte escolar.** Este requisito especifica que o sistema deva possibilitar o cadastro, a atualização, a remoção, a busca, e a visualização dos estudantes, das escolas, dos motoristas, da composição da frota, dos fornecedores e das rotas.
- **As funcionalidades de gestão devem ser possíveis de serem utilizadas mesmo com dados incompletos.** Este requisito aborda os entraves de necessitar de todos os dados para o uso do sistema. Deseja-se que o sistema opere, mesmo de forma limitada, com os dados existentes. O requisito tem impacto direto no modelo do banco de dados, uma vez que remove diversas obrigatoriedades no cadastro das entidades do transporte escolar.

- **O sistema deve ser capaz de importar e exportar dados.** O objetivo deste requisito é possibilitar que os usuários possam importar dados preexistentes, *e.g.*, de planilhas ou oriundos do censo. Além disso, deseja permitir que os dados cadastrados possam ser exportados para serem utilizados por outros sistemas.
- **Realizar a roteirização rapidamente.** Aqui, deseja-se que o sistema seja capaz de sugerir as melhores rotas com base nos dados cadastrados em poucos segundos. Sabe-se que o problema de roteirização do transporte escolar é classificado como NP-difícil, como consequência disso, não se conhece na literatura um algoritmo determinístico que execute em tempo polinomial para a solução do mesmo (PARK; KIM, 2010). Desta forma, a comunidade acadêmica investigou métodos baseados em meta-heurísticas como meio de aproximação da solução ótimo. Entretanto, os métodos podem requerer um tempo significativo para convergirem a um resultado (ELLEGOOD *et al.*, 2019). Assim, recomenda-se utilizar métodos exatos, mesmo que produzam resultados ligeiramente inferiores, mas que sejam capazes de sugerir as rotas em poucos segundos em computadores convencionais.
- **Acessar relatórios e estatísticas dos diferentes atores do transporte escolar.** Este requisito, visa possibilita que gestores possam encontrar índices e estatísticas a respeito dos atores cadastrados, tais como, taxa de ocupação de veículo, número de alunos atendidos, maior e menor quilometragem de rotas.
- **O sistema deve ser capaz de operar sem Internet.** Este requisito visa lidar com os problemas inerentes a instabilidade da conexão com a Internet encontrados em diversos municípios. Aqui, o *software* SETE deve ser capaz de operar, mesmo que de forma limitada, sem conectividade. As operações feitas durante esse estado devem ser salvas de modo a garantir a consistência do sistema.
- **Utilização apenas de *software* livre.** Com intuito de evitar onerar os municípios, o sistema deve ser projetado de maneira que não somente o código seja distribuído sob uma licença livre e de forma gratuita, mas que o mesmo não possua nenhuma dependência com bibliotecas proprietárias. Ao fazer isso, deseja-se possibilitar que os municípios não tenham de licenciar outros sistemas ao adotar o SETE.

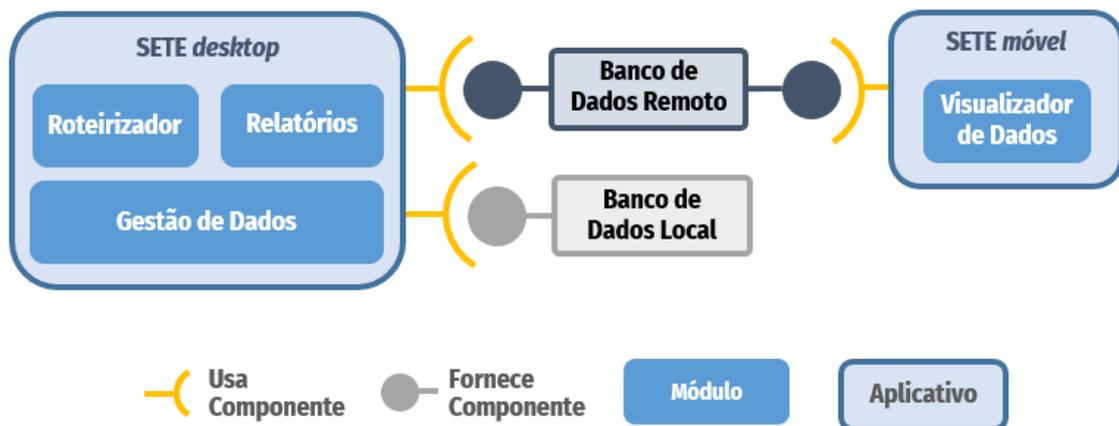
2.2. Arquitetura do Software

Tomando como base os requisitos especificados, propõe-se uma arquitetura de *software* modular e híbrida, conforme ilustrada na Figura 2.1. Note a presença de dois bancos de dados, um local e outro remoto. Esse arranjo é feito para possibilitar que o acesso e as operações computacionais sejam realizados localmente. Essa decisão arquitetural visa permitir que o sistema possa operar mesmo na ausência de conectividade com a Internet. Além disso, ao realizar as operações localmente, o sistema se torna mais responsivo, uma vez que não precisa de esperar a comunicação com a rede.

A arquitetura apresenta uma estrutura híbrida, tendo uma variação para o modo *desktop* e outra para o modo móvel. A interação entre as duas é feita utilizando o banco de dados remoto, situado em um servidor na nuvem. Tal banco tem como objetivo espelhar o banco de dados local utilizado pelo usuário no modo *desktop*. Ao fazer isso, é possível permitir o acesso aos dados locais em diferentes plataformas, como a partir do dispositivo móvel e outros sistemas.

A arquitetura do software é composta por vários módulos, unidades computacionais que encapsulam funções em comum do sistema. As unidades modulares interagem entre si para prover as funcionalidades do *software*. Por exemplo, o módulo de relatórios interage com o de gestão de dados para poder construir os indicadores e gráficos.

Figura 2.1: Arquitetura do *software* SETE



Fonte: Elaborada pelos autores.

Cada módulo tem como objetivo abordar um ou mais requisitos descritos, *e.g.*, o módulo de roteirização visa possibilitar sugerir rotas rapidamente para os usuários. Desta maneira, cada módulo realiza as seguintes atividades:

- **Gestão de Dados:** fornece ferramentas para cadastrar, remover, atualizar e buscar os dados referentes aos diferentes atores do transporte escolar (estudantes, escolas, frota, fornecedores e rotas). Nesse processo é feita uma validação extensivo dos dados de entrada, no intuito de garantir a confiabilidade dos mesmos. Além disso, o módulo visa possibilitar a importação e exportação de tais dados no sistema.
- **Roteirização:** utiliza algoritmos exatos para realizar a roteirização de veículos da frota. Também fornece mecanismos para parametrizar a execução do algoritmo. Por exemplo, permite que o gestor possa definir o tamanho máximo da rota a ser sugerida e a capacidade máxima dos veículos.
- **Relatórios:** fornece indicadores e descrição estatística dos dados dos diferentes atores do transporte escolar. Parte dos indicadores e das relações estatísticas foram extraídas dos utilizados para caracterizar o transporte escolar (FNDE/CEFTRU, 2017b).
- **Visualizador de dados:** fornece mecanismos de apresentar os dados do banco de dados local do sistema no dispositivo móvel. Para isso, utiliza o espelho dos dados presentes na nuvem.

2.3. Framework

As boas práticas de engenharia de *software* recomendam a utilização de *frameworks* existentes para acelerar o processo de construção de sistemas de informação e diminuir o número de problemas durante a concepção do mesmo (CERVANTES; KAZMAN, 2016; GHEZZI; JAZAYERI; MANDRIOLI, 2002). Um *framework* provê um esqueleto para a construção de um *software*, por exemplo, fornecendo rotinas para construção de janelas, para tratamento de ações em botões e impressão de arquivos. Pode-se então focar no conteúdo específico do sistema, por exemplo, nas ações que devem ser realizadas ao clicar em um botão ao invés das rotinas específicas para criar uma janela na tela do usuário.

Entre as diversas opções existentes, adotou-se o *framework* Electron (GITHUB, 2019), que possibilita a utilização de linguagens e bibliotecas de programação Web, como Javascript e NodeJS, para construção de sistemas *desktop* e multiplataformas, isto é, que podem ser executados em diferentes sistemas operacionais.

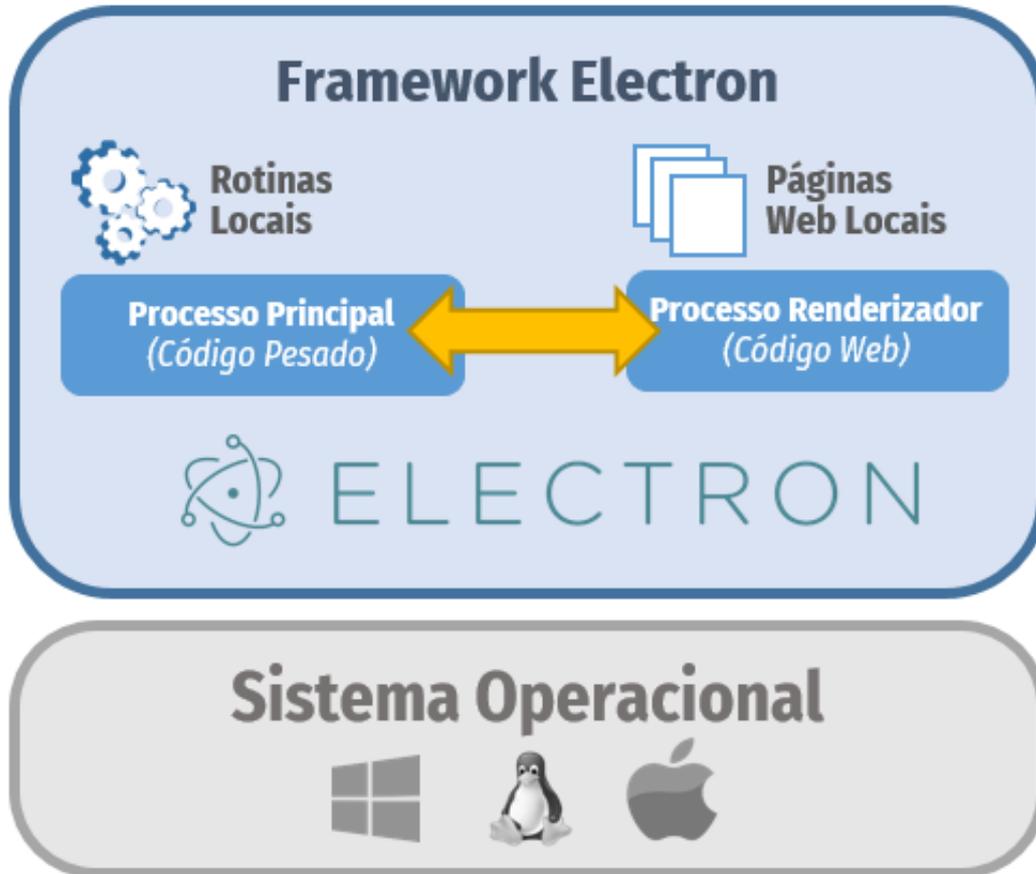
A decisão por esse *framework* se deu por vários fatores. Primeiro, por ser um *framework* voltado a sistemas *desktops*, que executam localmente na máquina do usuário, é possível programar mecanismos para lidar com a instabilidade recorrente da conexão com a Internet que se encontra presente em vários municípios brasileiros. Segundo, ao possibilitar a utilização de tecnologias Web diretamente no *desktop*, pode-se beneficiar da rapidez e flexibilidade das mesmas para a criação de interfaces (JENSEN, 2017). Terceiro, ao optar pelo Electron, pode-se posteriormente migrar o código do SETE facilmente para uma versão Web, uma vez que o código do sistema está descrito utilizando tecnologias nativas da Web, como Javascript, NodeJS, CSS e HTML. Por fim, o *framework* também está disponibilizado sem custos sob a licença de *software* livre MIT (OSI, 2019a), a mesma utilizada pelo SETE.

Para possibilitar a execução de tecnologias Web em modo *desktop*, o *framework* Electron utiliza dois processos distintos, *main* (principal) e *renderer* (renderizador), como ilustrado na Figura 2.2. O processo renderizador é responsável por executar o código da interface do sistema, ou seja, concentra o código referente a interação com o usuário. Já o processo principal é destinado a execução de algoritmos mais complexos, tais como a roteirização e operações com o banco de dados. Além disso, o processo principal é o principal meio de interagir com o sistema operacional subjacente permitindo acessar recursos do mesmo, *e.g.*, impressoras cadastradas e arquivos no disco rígido.

A separação de processos é importante pois permite aumentar o desempenho do sistema. Por exemplo, caso o código principal e de interface coexistissem dentro de um único processo, o sistema poderia apresentar sinais de lentidão quando o usuário clicasse em um item da interface (*e.g.*, um botão para salvar um dado ou gerar uma rota). O motivo para isso é que o sistema ficaria bloqueado aguardando o término da execução da rotina. Ao separar as tarefas em dois processos, o *framework* possibilita que o código pesado em pano de fundo, isto é, em paralelo e independentemente daquele relacionado a interação

com o usuário. Por essa razão, o *software* SETE segue o mesmo princípio de separação de tarefas do Electron, fazendo uso dos dois processos para distribuição do código.

Figura 2.2: Estrutura do *framework* Electron



Fonte: Elaborada pelos autores.

2.4. Tecnologias de Informação Geográfica

O *software* SETE possui um conjunto de funcionalidades semelhantes a um Sistema de Informação Geográfica (os *Geographic Information Systems* – GIS), como a capacidade de gerir e visualizar espacialmente os atores (estudantes, escolas, *etc*) e sugerir rotas automaticamente. Para prover isso, primeiramente é necessário utilizar uma base de dados de mapas. Nesse contexto, o SETE utiliza a suíte de mapas colaborativos fornecidos pelo projeto OpenStreetMap (NEIS; ZIELSTRA, 2014). Esta base é disponível gratuitamente sob a licença livre *ODbL* (OPEN KNOWLEDGE FOUNDATION, 2019).

Os mapas do OpenStreetMaps são construídos de formas colaborativas e seus dados podem ser apresentados tanto em formato vetorial, quanto por imagens de satélites, que são fornecidas gratuitamente pela plataforma Microsoft Bing (COAST, 2019; HAKLAY; WEBER, 2008). A Figura 2.3 ilustra um exemplo de mapa do OpenStreetMap dentro do SETE no módulo de dados. Na tela de cadastro de um estudante, o usuário pode utilizar a visão de satélite do mapa para apontar a posição do mesmo.

A manipulação dos dados do OpenStreetMaps é feita através da biblioteca de *software* livre OpenLayers (FARKAS, 2016; METACARTA, 2019). A mesma é disponibilizada sob a licença de *software* livre de duas cláusulas BSD (OSI, 2019b). O SETE utiliza a biblioteca para apresentar itens na tela, *e.g.*, mostrar a localização geográfica das escolas. Também se utiliza a biblioteca para captura a interação dos usuários com os elementos do mapa. Por exemplo, o *software* SETE captura as ações de desenho na tela e transforma as mesmas nos traçados das rotas.

Figura 2.3: Exemplo de utilização da tecnologia OpenStreetMaps para cadastro da localização de usuários.

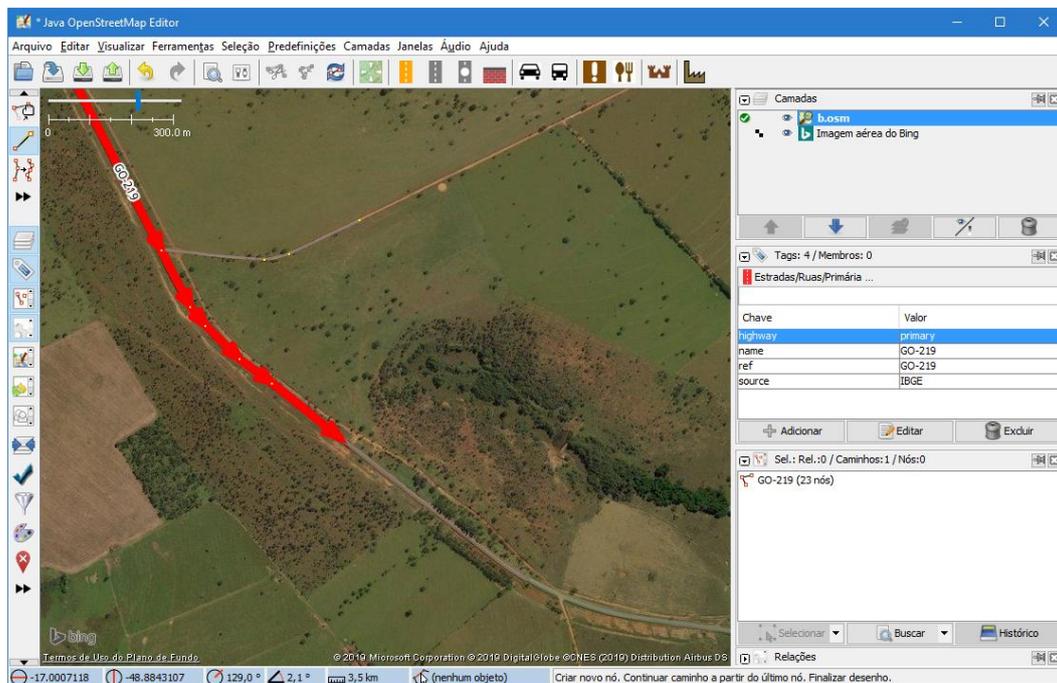


Fonte: Elaborada pelos autores.

O projeto OpenStreetMaps também é utilizado para auxiliar nas etapas de roteirização. Especificamente, um problema recorrente é a dificuldade de obtenção da malha viária rural para a execução dos algoritmos (GRASER; STRAUB; DRAGASCHNIG, 2015; MOREIRA; JÁCOMO; GRIEBELER, 2016). Precisamente, para possibilitar a geração de rotas, os usuários devem completar as vias que faltam da malha. Para possibilitar que os mesmos possam fazer isso, criou-se uma adaptação do *software* de edição de malhas Java OpenStreetMap Editor (STÖCKER, 2019) chamada S2EM (Sistema Eletrônico de Edição de Malha). O *software* resultante é distribuído sob a licença GPL v2 (OSI, 2019c) e seu instalador vêm em conjunto com o SETE.

Através do *software* livre de edição de malhas S2EM é possível complementar as vias que faltam utilizando uma interface gráfica, como ilustrado na Figura 2.4. As vias podem ser salvas em um arquivo que posteriormente é utilizado com entrada para o algoritmo de roteirização. Além disso, deve ser ressaltado que, como as edições são feitas tomando como base os mapas do OpenStreetMaps, é possível que as mesmas sejam enviadas para a base colaborativa do projeto, possibilitando assim uma retroalimentação e uma melhoria global da qualidade do mapa para todos os usuários.

Figura 2.4: Exemplo de utilização do *software* livre de edição de malha S2EM.



Fonte: Elaborada pelos autores.

Note que os dados são os elementos em comum entre todos os módulos mencionados. Nesse sentido, a próxima seção apresente como os mesmos estão organizados no sistema e em seguida, é feito um detalhamento de como as funções de cada módulo são realizadas e das interações entre eles.

3. Gestão de Dados

Os dados são os elementos centrais de um sistema de informação (SILBERSCHATZ; KORTH; SUDARSHAN, 2006). No contexto do SETE, todos os módulos interagem diretamente com os dados do sistema. Assim, faz-se necessidade ter uma definição dos dados e de seu armazenamento para possibilitar a interação e compreensão dos mesmos tanto pelos módulos do *software*, quanto por sistemas externos que venham utilizar os dados do SETE. A construção do banco de dados inicia-se pelo modelo conceitual, onde é feito a identificação dos atores e das relações entre eles. Em seguida, detalha-se os atributos de cada um dos atores descritos na construção do modelo lógico do banco de dados. Por fim, é feita a implementação do mesmo em um sistema gerenciador de banco de dados gerando assim o modelo físico dos dados. Desta forma, esse capítulo descreve o modelo de dados seguindo as etapas descritas.

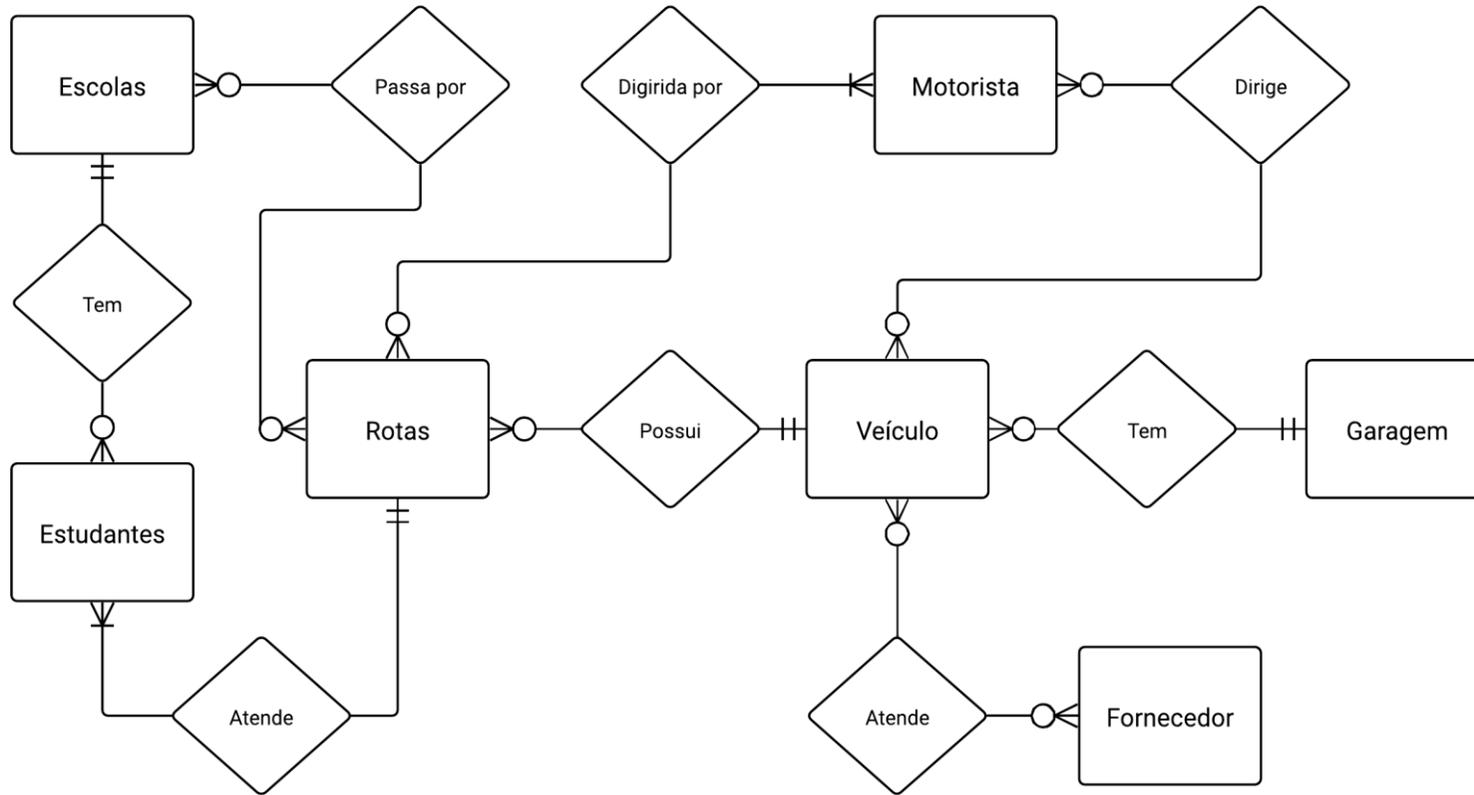
3.1. Modelo Conceitual

O modelo conceitual visa identificar a partir dos requisitos levantados os principais atores que são necessários para construir o sistema (BATINI; CERI; NAVATHE, 1992). Esse modelo é descrito em alto nível de abstração, isto é, o importante é capturar as relações entre os atores e não o detalhamento dos mesmos.

Considerando o contexto do transporte escolar e os requisitos descritos, identifica-se os seguintes atores: estudantes, ponto de paradas, escolas, motoristas, veículos, garagem, fornecedores e rotas. A Figura 3.1 ilustra o modelo conceitual envolvendo esses atores. Os retângulos representam as entidades do sistema SETE, enquanto que o losango indica o relacionamento entre elas. Por exemplo, a rota atende um ou mais estudantes. Para fazer isso, deve existir um vínculo na base de dados que relacione os dois atores.

Os traços nas arestas dos relacionamentos indicam a restrição entre as entidades, usando a notação de *crow's foot* (EVEREST, 1976). Utilizando a notação é possível identificar as cardinalidades dos relacionamentos. Por exemplo, a relação *tem* que vincula o ator escola com o estudante indica que uma escola pode ter zero ou mais alunos atendidos pelo transporte escolar, enquanto que um único estudante está matriculado em uma única escola, isto é, apresenta um e somente um vínculo por vez com uma escola.

Figura 3.1: Modelo conceitual do banco de dados do sistema.



Fonte: Elaborada pelos autores.

Além disso, no tocante aos atores descritos, observa-se que as rotas e os veículos são elementos centrais de interligação entre as diferentes entidades. Por exemplo, as rotas atendem estudantes, que por sua vez estão matriculados em uma dada escola. Relação semelhante para os veículos, que são dirigidos por motoristas e possuem um vínculo com uma garagem.

3.2. Modelo Lógico

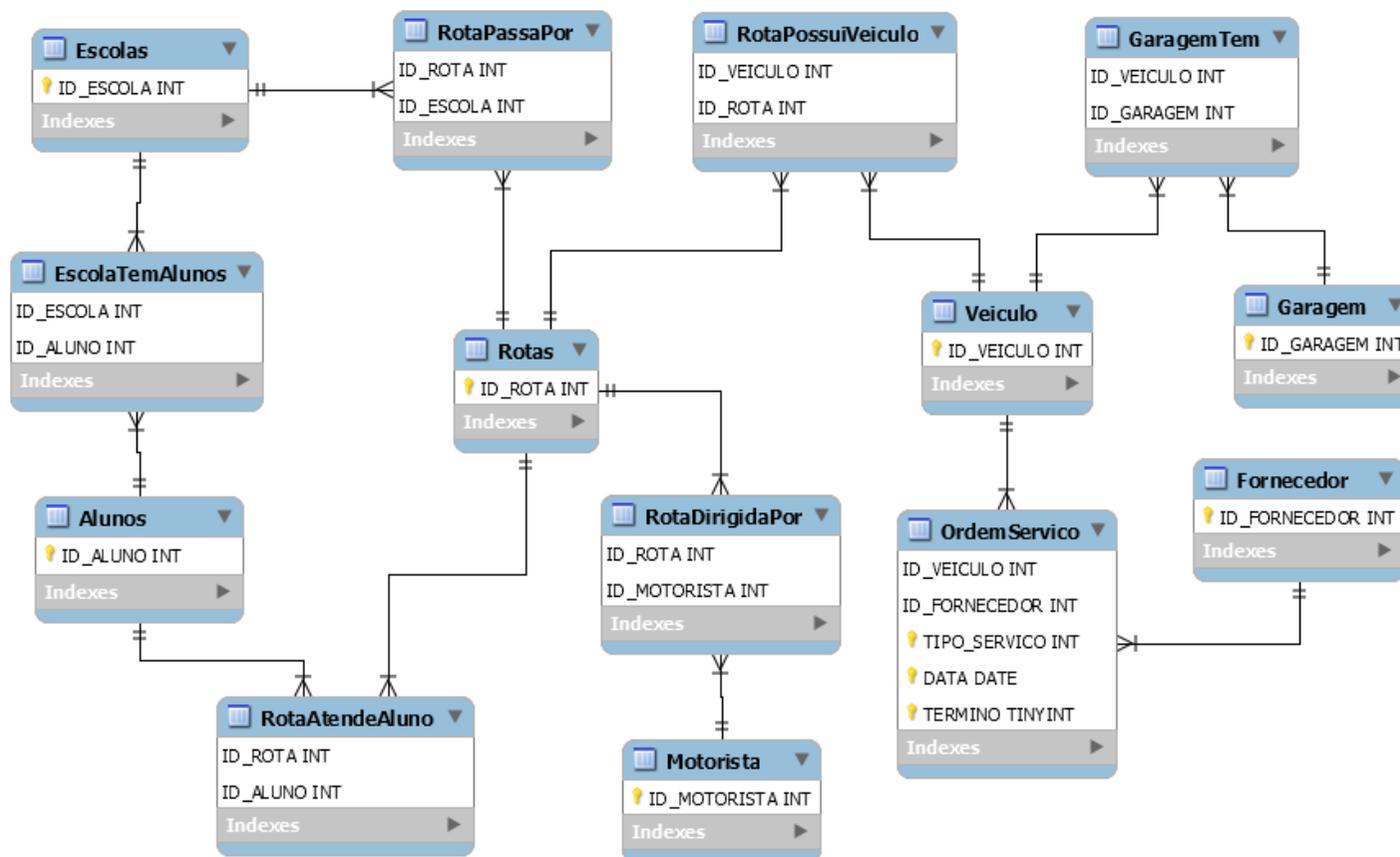
A partir do modelo conceitual, pode-se construir o modelo lógico do banco de dados. O modelo lógico visa detalhar o esquema atributos dos atores, isto é, o conjunto de variáveis utilizado para representar cada entidade (SILBERSCHATZ; KORTH; SUDARSHAN, 2006). Além disso, o modelo torna explícito o relacionamento entre as entidades através dos conceitos de chave primária e de chave estrangeira. O diagrama de entidade e relacionamento de Peter Chen (1976) é o modelo preferencial para descrever o modelo lógico (DATE, 2003).

A Figura 3.2 ilustra o modelo lógico (simplificado) de banco de dados utilizando apenas as chaves primárias e estrangeiras dos atores descritos no modelo conceitual. Os atributos que possuem uma chave dourada no diagrama representam uma chave primária, enquanto que aqueles que não possuem um ícone, presentes nas relações, representam uma chave estrangeira.

A chave primária é utilizada para identificar unicamente um dado dentro do sistema. Já a chave estrangeira permite referenciar uma chave primária de outro ator. Utilizou-se a chave estrangeira para criar um relacionamento entre dados de diferentes entidades. Por exemplo, a relação *RotaDirigidaPor* vincula uma rota a um ou mais motoristas, utilizando as chaves primárias ID_ROTA (Rotas) e ID_MOTORISTA (Motorista). A partir desses relacionamentos pode-se utilizar consultas para descobrir a ligação entre os atores. Por exemplo, pode-se correlacionar as relações *RotaAtendeAluno* e *RotaDirigidaPor* para descobrir quais motoristas atendem um determinado grupo de alunos.

Optou-se por limitar a descrição dos atores no diagrama a somente as chaves primárias e estrangeiras para destacar o relacionamento das entidades. Nas próximas subseções são descritos os atributos detalhados de cada um dos atores.

Figura 3.2: Modelo lógico (e simplificado) do banco de dados do sistema.



Fonte: Elaborada pelos autores.

3.2.1. Escolas

A modelagem da entidade escola no sistema tentou utilizar campos pré-existentes em outras fontes de dados, tais como do censo escolar (INEP, 2018). A Tabela 3.1 descreve o esquema de dado utilizado e as respectivas restrições:

Tabela 3.1 Esquema de dados da entidade Escola.

Coluna	Tipo de dado	Restrições	Descrição
ID_ESCOLA	INT	Chave Primária Único Não Nulo	Identificador do dado
MEC_CO_UF	INT	Não Nulo	UF da Escola
MEC_CO_MUNICIPIO	INT	Não Nulo	Município da Escola
MEC_NO_ENTIDADE	VARCHAR	Não Nulo	Nome da Escola
MEC_TP_DEPENDENCIA	INT	Não Nulo	Tipo de dep. da escola: 1. Federal 2. Estadual 3. Municipal 4. Privada
MEC_TP_LOCALIZACAO	INT	Não Nulo	Localização da escola: 1. Urbana 2. Rural
MEC_TP_LOCALIZACAO_DIFERENCIADA	INT		Escola está localizada em área diferenciada? 0: Não 1: Assentamento 2: Terra Indígena 3: Quilombo
MEC_IN_REGULAR	BOOLEAN	Falso por padrão	Escola oferta ensino regular?
MEC_IN_EJA	BOOLEAN	Falso por padrão	Escola oferta ensino para jovens adultos?
MEC_IN_PROFISSIONALIZANTE	BOOLEAN	Falso por padrão	Escola oferta ensino profissionalizante?
LOC_LATITUDE	REAL		Latitude da escola
LOC_LONGITUDE	REAL		Longitude da escola
LOC_CEP	VARCHAR		CEP da escola
LOC_ENDERECO	VARCHAR		Endereço da escola
CONTATO_RESPONSAVEL	VARCHAR		
CONTATO_TELEFONE	VARCHAR		
HORARIO_MATUTINO	BOOLEAN	Falso por padrão	Escola funciona no período matutino?
HORARIO_VESPERTINO	BOOLEAN	Falso por padrão	Escola funciona no período vespertino?
HORARIO_NOTURNO	BOOLEAN	Falso por padrão	Escola funciona no período noturno?
ENSINO_SUPERIOR	BOOLEAN	Falso por padrão	Escola oferta ensino superior?
ENSINO_MEDIO	BOOLEAN	Falso por padrão	Escola oferta ensino médio?
ENSINO_FUNDAMENTAL	BOOLEAN	Falso por padrão	Escola oferta ensino básico?

Campo obrigatório Campo não obrigatório

Os atributos que começam com o prefixo “MEC” são oriundos do dicionário de metadados do censo escolar. A decisão pela utilização dos mesmos campos no sistema SETE, incluindo tipo e nome de atributo, foi feita com intuito de manter a compatibilidade com o censo e, conseqüentemente, facilitar a importação de dados no *software*.

As variáveis do tipo *boolean* representam variáveis lógicas, que podem assumir um valor verdadeiro ou falso, que por sua vez são mapeados para os valores 1 e 0 respectivamente. Por exemplo, a variável booleana MEC_IN_EJA indica se a escola em questão possui ou não ensino voltado a jovens e adultos (EJA). Por padrão, caso seja omitido tal informação, o valor da variável será considerado como falso (0), isto é, a mesma não oferece essa modalidade de ensino. Entretanto, no cadastro se for indicado que a escola realiza esse tipo de ensino, a mesma será verdadeira e terá como valor o inteiro 1.

Observe que apesar da grande quantidade de atributos, apenas uma pequena parte é dita obrigatória. Como obrigatório se tem apenas dados básicos, tais como nome, modalidade de ensino oferecido, cidade e estado em que a escola se encontra. Note que é importante saber a cidade da escola, uma vez que pode ocorrer de alunos de uma cidade estudarem em escolas localizadas em outros municípios, *e.g.*, o transporte de alunos universitários.

Grande parte dos dados no cadastro da escola são não obrigatórios, dentre eles têm se o par de localização latitude e longitude, o CEP e o endereço. A opção por tornar tais campos em não obrigatório possibilita que mesmo um município que não possua ou saiba informar tais dados continue utilizando, mesmo que de forma limitada, o *software*. Obviamente a ausência de informações georeferenciadas impossibilita a utilização de módulos mais avançados, tais como a roteirização.

3.2.2. Estudantes

Semelhante as escolas, tentou-se fazer um reaproveitamento do dicionário de dados do censo escolar para os estudantes. Entretanto, como o censo escolar não divulga os dados diretos dos discentes, como nome e endereço, o reuso de variáveis desta fonte foi limitado. Apesar disso, reutilizou-se as variáveis referentes aos metadados, como presença ou não de deficiência e localização do aluno. A Tabela 3.2 apresenta o dicionário de dados dos alunos utilizados no *software*.

Tabela 3.2 Esquema de dados da entidade Estudante.

Coluna	Tipo de dado	Restrições	Descrição
ID_ALUNO	INT	Chave Primária Único Não Nulo	Identificador do dado
NOME	VARCHAR	Não Nulo	Nome do aluno
DATA_NASCIMENTO	DATE	Não Nulo	Data de nascimento do aluno
SEXO	INT	Não Nulo	Sexo do aluno: 1: Masculino 2: Feminino 3: Não informado
COR	INT	Não Nulo	Cor/raça do aluno: 1: Amarelo 2: Branco 3: Indígena 4: Pardo 5: Preto
TURNO	INT	Não Nulo	Turno de estudo do aluno: 1: Matutino 2: Vespertino 3: Integral 4: Noturno
NIVEL	INT	Não Nulo	Nível do aluno: 1: Infantil 2: Fundamental 3: Médio 4: Superior 5: Outro
NOME_RESPONSAVEL	VARCHAR	Não Nulo	Nome do responsável
GRAU_RESPONSAVEL	INT	Não Nulo	Grau de parentesco: 0: Pai, Mãe, Padrasto ou Madrasta 1: Avô ou Avó 2: Irmão ou Irmã 4: Outro Parente
MEC_TP_LOCALIZACAO	INT	Não Nulo	Localização do aluno: 1. Urbana 2. Rural
CPF	INT		CPF do aluno
TELEFONE_RESPONSAVEL	INT		Telefone do responsável
LOC_LATITUDE	REAL		Latitude da escola
LOC_LONGITUDE	REAL		Longitude da escola
LOC_CEP	VARCHAR		CEP da escola
LOC_ENDERECO	VARCHAR		Endereço da escola
DA_PORTEIRA	BOOLEAN	Falso por padrão	Tem porteira?
DA_MATABURRO	BOOLEAN	Falso por padrão	Tem mata-burro?
DA_COLCHETE	BOOLEAN	Falso por padrão	Tem colchete?
DA_ATOLEIRO	BOOLEAN	Falso por padrão	Tem atoleiro?

DA_PONTERUSTICA	BOOLEAN	Falso por padrão	Tem ponte rústica?
DEF_CAMINHAR	BOOLEAN	Falso por padrão	Tem deficiência física?
DEF_OUVIR	BOOLEAN	Falso por padrão	Tem deficiência auditiva?
DEF_ENXERGAR	BOOLEAN	Falso por padrão	Tem deficiência visual?
DEF_MENTAL	BOOLEAN	Falso por padrão	Tem deficiência mental?

Campo obrigatório Campo não obrigatório

Semelhante as escolas, existe pouca obrigatoriedade na entrada de dados dos estudantes. Especificamente, exige-se a obrigatoriedade de informar alguns dados pessoais, tais como nome, cor e data de nascimento, e dados de ensino, como turno e nível do discente. Além disso, pede-se o nome do responsável e o respectivo grau de parentesco. Isto é feito para possibilitar que o gestor possa recorrer ao responsável em caso de problemas.

As variáveis lógicas com o prefixo “DA” indicam a presença ou não de dificuldades de acesso na residência do aluno, isto é, de impedâncias na localização dos mesmos. Essa informação é importante pois influencia tanto no planejamento, quanto na operação das rotas. Por exemplo, uma rota que atende um aluno cujo acesso possui um colchete pode sofrer um atraso ou requerer um traçado diferente.

De maneira análoga, as variáveis com o prefixo “DEF” expressam a presença ou não de um tipo de deficiência. Observe que tanto as variáveis de impedância, quanto as que relatam uma presença de deficiência são não obrigatórias.

As variáveis de localização do discente começam com o prefixo “LOC”. Por padrão não é necessário georeferenciar o aluno. Entretanto, ao não realizar essa etapa, não será possível utilizar outras ferramentas do *software* SETE, como a geração automática de rotas e os indicadores geoespaciais.

3.2.3. Motoristas

A modelagem dos motoristas no sistema teve como objetivo obter variáveis relacionadas diretamente a atividade de transporte. Desta forma, além dos dados básicos, o foco situa em representar os dados de habilitação, como número e nível de carta de motorista e horário de trabalho. Além disso, por questões de segurança, é destinado um campo para envio do número e do documento em si de antecedentes criminais. Mesmo como optativo,

a utilização desse campo pode ser útil para garantir a segurança no transporte. A Tabela 3.3 descreve o esquema completo de dados.

Tabela 3.3 Esquema de dados da entidade Motorista.

Coluna	Tipo de dado	Restrições	Descrição
ID_MOTORISTA	INT	Chave Primária Único Não Nulo	Identificador do dado
NOME	VARCHAR	Não Nulo	Nome do aluno
DATA_NASCIMENTO	DATE	Não Nulo	Data de nascimento do aluno
SEXO	INT	Não Nulo	Sexo do aluno: 1: Masculino 2: Feminino 3: Não informado
CNH	INT	Não Nulo	Número da CNH
CPF	INT		CPF do motorista
TELEFONE	INT		Telefone do motorista
ANT_CRIMINAIS	VARCHAR		Número do antecedentes criminais
ARQUIVO_DOCPESSOAIS_ANEXO	VARCHAR		Arquivo de documentos pessoais
TURNO_MANHA	BOOLEAN	Falso por padrão	Trabalha no turno da manhã?
TURNO_TARDE	BOOLEAN	Falso por padrão	Trabalha no turno vespertino?
TURNO_NOITE	BOOLEAN	Falso por padrão	Trabalha no turno noturno?
TEM_CNH_A	BOOLEAN	Falso por padrão	Tem CNH categoria A?
TEM_CNH_B	BOOLEAN	Falso por padrão	Tem CNH categoria B?
TEM_CNH_C	BOOLEAN	Falso por padrão	Tem CNH categoria C?
TEM_CNH_D	BOOLEAN	Falso por padrão	Tem CNH categoria D?
TEM_CNH_E	BOOLEAN	Falso por padrão	Tem CNH categoria E?

Campo obrigatório Campo não obrigatório

As variáveis lógicas com prefixo “TEM” indica se o motorista possui ou não habilitação para manusear um veículo com a devida categoria. Já aquelas iniciadas com “TURNO” expressam o período em que o motorista está disponível para conduzir o veículo.

Destaca-se a variável opcional ANT_CRIMINAIS que expressa o número do documento de antecedentes criminais. O gestor também pode anexar tal arquivo, ao fazer isso o conteúdo do mesmo é armazenado na variável ARQUIVO_DOCPESSOAIS_ANEXO.

3.2.4. Veículos

Os veículos são elementos centrais no *software*, uma vez que possui relacionamento com diversos elementos do modelo. No contexto do transporte escolar, é importante possuir os dados de algumas características básicas do mesmo, como modelo e placa, e atributos relacionados a operação, como capacidade e quilometragem (se aplicável). Com base nisso, descreveu-se o modelo A Tabela 3.4 descreve o esquema completo de dados.

Tabela 3.4 Esquema de dados da entidade Veículo.

Coluna	Tipo de dado	Restrições	Descrição
ID_VEICULO	INT	Chave Primária Único Não Nulo	Identificador do dado
PLACA	VARCHAR	Não Nulo	Placa do veículo
ANO	DATE	Não Nulo	Data de produção do veículo
RENAVAM	VARCHAR	Não Nulo	RENAVAM do veículo
ORIGEM	INT	Não Nulo	Origem do veículo: 1: Próprio 2: Terceirizado
MODO	INT	Não Nulo	Modalidade do veículo: 1: Rodoviário 2: Aquaviário
TIPO	INT	Não Nulo	Tipo do veículo: 1: Ônibus 2: Micro-ônibus 3: Van 4: Kombi 5: Caminhão 6: Caminhonete 7: Motocicleta 8: Animal de tração 9: Lancha/Voadeira 10: Barco de madeira 11: Barco de alumínio 12: Canoa motorizada 13: Canoa a remo
CAPACIDADE	INT	Não Nulo	Número de assentos no veículo

MODELO	INT		Modelo do veículo: 1: Não se aplica 2: ORE 1 3: ORE 1 (4x4) 4: ORE 2 5: ORE 3 6: ORE 4 7: ONUREA 8: Lancha a Gasolina 9: Lancha a Diesel
KM_INICIAL	REAL		Quilometragem do veículo no momento do cadastro
KM_ATUAL	REAL		Quilometragem do veículo atualmente.
MANUTENCAO	BOOLEAN	Falso por padrão	Veículo se encontra em manutenção?

Campo obrigatório Campo não obrigatório

Em termos dos dados de entrada obrigatórios, destaca-se o campo **CAPACIDADE**, que indica a quantidade máxima de assentos que o veículo comporta. Observe que não existe um campo explícito para a capacidade atual, uma vez que a mesma depende do horário em que o veículo está sendo utilizado, *e.g.*, a capacidade no período da manhã pode ser diferente do período da tarde. Para computar a capacidade atual pode-se correlacionar os dados das relações *RotaPossuiVeiculo* e *RotaAtendeAluno* para obter a quantidade de alunos atendidos e, conseqüentemente, calcular a lotação.

Os campos **PLACA** e **RENAVAM** possibilitam descrever os dados cadastrais básicos do veículo. Como tais dados são necessários para operar o veículo, decidiu-se tornar ambos dados obrigatórios no modelo de dados do *software*.

O campo **TIPO** permite explicitar a categoria do veículo de transporte escolar. Caso o veículo tenha sido adquirido dentro do programa Caminho da Escola, pode-se detalhar o modelo obtido através das opções do campo **MODELO**.

Por fim, o campo **KM_INICIAL** e **KM_ATUAL** especifica a quilometragem no ato de cadastro e a atual do veículo. O campo contendo a quilometragem atual é atualizado a cada manutenção. Utilizando tal atributo, em conjunto com o intervalo de tempo, pode-se capturar o uso do veículo.

3.2.5. Rotas

Conforme ilustrado no modelo lógico e conceitual, as rotas são os elementos com maior interconexão no sistema. Cada rota interage diretamente com quatro atores: estudantes, escolas, veículos e motoristas. Além disso, interagem com as restantes de forma indireta.

No contexto da gestão do transporte escolar, destaca-se algumas características das rotas, tais como alunos e escolas atendidas. Observe que esses campos são extraídos a partir das relações descritas, *e.g.*, *RotaAtendeAluno* e *RotaPassaPor* respectivamente.

Em relação as características individuais das rotas, destaca-se a necessidade de conhecer a quilometragem, tempo, traçado, *etc.* Desta forma, o sistema utiliza o esquema detalhado na Tabela 3.5 para representar as rotas.

Tabela 3.5 Esquema de dados da entidade Rota.

Coluna	Tipo de dado	Restrições	Descrição
ID_ROTA	INT	Chave Primária Único Não Nulo	Identificador do dado
NOME	VARCHAR	Não Nulo	Nome da rota
KM	REAL	Não Nulo	Tamanho da rota (em km)
TEMPO	INT	Não Nulo	Tempo da rota (em min)
TIPO	INT	Não Nulo	Origem do veículo: 1: Rodoviária 2: Aquaviária 3: Mista
SHAPE	VARCHAR		Descrição do traçado da rota de forma vetorial.
HORA_IDA_INICIO	DATE		Horário de início da viagem de ida da rota
HORA_IDA_TERMINO	DATE		Horário de término da viagem de ida da rota
HORA_VOLTA_INICIO	DATE		Horário de início da viagem de volta da rota
HORA_VOLTA_TERMINO	DATE		Horário de término da viagem de volta da rota
DA_PORTEIRA	BOOLEAN	Falso por padrão	Passa por porteira?
DA_MATABURRO	BOOLEAN	Falso por padrão	Passa por mata-burro?
DA_COLCHETE	BOOLEAN	Falso por padrão	Passa por colchete?
DA_ATOLEIRO	BOOLEAN	Falso por padrão	Passa por atoleiro?

TURNO_MATUTINO	BOOLEAN	Falso por padrão	Funciona no período matutino?
TURNO_VESPERTINO	BOOLEAN	Falso por padrão	Funciona no período vespertino?
TURNO_NOTURNO	BOOLEAN	Falso por padrão	Funciona no período noturno?

Campo obrigatório Campo não obrigatório

Esse elemento possui poucos campos obrigatórios, visto a dificuldade dos gestores em conhecer os detalhes das rotas. O atributo NOME fornece um mecanismo mais amigável de identificar as rotas, *e.g.*, o gestor pode associar o campo as regiões que o veículo percorre. Já o campo KM e TEMPO especificam respectivamente o tamanho (em km) e o tempo de percurso da rota (em min).

O traçado da rota é descrito opcionalmente no campo SHAPE. Aqui, optou-se por utilizar o formato de dados espacial aberto GeoJSON (BUTLER *et al.*, 2016). O *software* provê um editor que permite desenhar as rotas e convertê-las para o formato GeoJSON. Além disso, as rotas geradas automaticamente, via roteirização, também são armazenadas neste formato. É válido destacar que o formato GeoJSON é um padrão livre e pode ser utilizado em diversos sistemas de informação geográfico (KRALIDIS, 2008).

Semelhante aos estudantes, as variáveis com prefixo “DA” indicam a presença de uma ou devida dificuldade de acesso na rota, como mata-burro, porteira, *etc.* Já os campos opcionais prefixados com HORA_IDA e HORA_VOLTA indicam os horários de saída e de chegada das viagens de ida e de volta respectivamente. Por fim, os campos lógicos prefixados por TURNO indicam se a rota atende no turno especificado.

3.2.6. Garagem

A garagem armazena um ou mais veículos. Devido ao papel restrito, é uma entidade relativamente simples e seus dados se limitam ao identificador e localização (opcional) da mesma. A descrição completa da entidade pode ser vista na Tabela 3.6.

Tabela 3.6 Esquema de dados da entidade Garagem.

Coluna	Tipo de dado	Restrições	Descrição
ID_GARAGEM	INT	Chave Primária Único Não Nulo	Identificador do dado

LOC_LATITUDE	REAL		Latitude da garagem
LOC_LONGITUDE	REAL		Longitude da garagem
LOC_ENDERECO	VARCHAR		Endereço da garagem
LOC_CEP	VARCHAR		CEP da garagem

Campo obrigatório Campo não obrigatório

3.2.7. Fornecedor

No contexto do *software* SETE, a entidade fornecedora representa um ator que realiza serviços voltado aos veículos, como provimento de combustível manutenção de peças. Desta forma, utiliza-se como atributos desse elemento dados comuns a empresas, como CNPJ e nome fantasia, além de informações a respeito do ramo de atuação. A descrição completa da entidade é ilustrada na Tabela 3.7.

Tabela 3.7 Esquema de dados da entidade Fornecedor.

Coluna	Tipo de dado	Restrições	Descrição
ID_FORNECEDOR	INT	Chave Primária Único Não Nulo	Identificador do dado
CNPJ	VARCHAR		CNPJ do fornecedor
NOME	VARCHAR		Nome fantasia do fornecedor
RAMO_MECANICA	BOOLEAN	Falso por padrão	Fornecedor atua no ramo de mecânica?
RAMO_COMBUSIVEL	BOOLEAN	Falso por padrão	Fornecedor atua no ramo de combustível e lubrificantes?
RAMO_SEGURO	BOOLEAN	Falso por padrão	Fornecedor atua no ramo de seguros?
LOC_LATITUDE	REAL		Latitude do fornecedor
LOC_LONGITUDE	REAL		Longitude do fornecedor
LOC_ENDERECO	VARCHAR		Endereço do fornecedor
LOC_CEP	VARCHAR		CEP da fornecedor

Campo obrigatório Campo não obrigatório

Além dos campos básicos, como CNPJ e NOME destaca-se a utilização dos campos lógicos prefixados com “RAMO”. Cada campo desse indica se o fornecedor atua ou não no ramo especificado.

Como dito, no *software* SETE o ator fornecedor atua diretamente com os veículos. Desta forma, com intuito de explicitar esse relacionamento, destaca-se na Tabela 3.8 a relação *OrdemDeServico* que vincula as duas entidades.

Tabela 3.8 Esquema de dados da entidade *OrdemDeServico*.

Coluna	Tipo de dado	Restrições	Descrição
ID_FORNECEDOR	INT	Chave Primária Único Não Nulo	Identificador do fornecedor
ID_VEICULO	INT	Chave Primária Único Não Nulo	Identificador do veículo
TERMINO	BOOLEAN	Chave Primária Não Nulo	O serviço já foi concluído?
DATA	DATE	Chave Primária Não Nulo	Data em que o serviço foi/será feito
TIPO_SERVICO	INT	Chave Primária Não Nulo	Tipo: 1: Combustível 2: Lubrificante 3: Pag. seguros 4: Man. Preventiva 5: Manutenção
COMENTARIO	VARCHAR		Comentário a respeito do serviço prestado

Campo obrigatório Campo não obrigatório

Observe que a relação descrita possui uma chave primária composto de cinco atributos: identificador dos fornecedores e veículo, data e tipo de serviço e um campo lógico para indicar se o mesmo foi concluído. A utilização dos dois identificadores vincula os atores, isto é, associa o serviço prestado pelo fornecedor a um veículo específico. Já o campo **TIPO_SERVICO** especifica a categoria do que foi ou será realizado pelo fornecedor, enquanto que **COMENTARIO** possibilita que o gestor inclua informações adicionais a respeito do trabalho.

O campo **TERMINO** permite indicar se o serviço já foi concluído. Ao combinar esse dado com a **DATA**, pode-se expressar um serviço agendado, por exemplo, uma manutenção ou o abastecimento do veículo.

3.3. Tecnologias de Armazenamento

As tecnologias de gerenciamento de banco de dados possuem uma importância crucial na concepção dos sistemas de informação (SILBERSCHATZ; KORTH; SUDARSHAN,

2006). Isto porque elas impactam diretamente no desempenho do sistema, uma vez que os mesmos possuem suas ações centradas na manipulação de dados. Devido a isso, tais tecnologias são comumente disponibilizadas como um *software* a parte, requerendo que o usuário faça uma instalação separada do sistema gerenciador de banco de dados.

Com intuito de evitar que o usuário final tenha de realizar tais tarefas, o SETE adotou a tecnologia de banco de dados SQLite (ALLEN; OWENS, 2010; DWAYNE RICHARD HIPP, 2019). O motivo para tal é que o SQLite é um sistema de gerenciamento de banco de dados que pode ser embarcado em outros *softwares*, *i.e.*, o mesmo fornece mecanismos para armazenar e manipular dados de forma portátil (sem requerer instalação prévia).

Para prover a portabilidade, o SQLite opera diretamente em arquivos textuais do sistema. A tecnologia fornece todas as operações presentes na linguagem de consulta *Structured Query Language* (SQL) (SILBERSCHATZ; KORTH; SUDARSHAN, 2006). Por essa razão o SQLite é uma das principais tecnologias de banco de dados em utilização, estando presente nos sistemas operacionais Mac e Windows e em aparelhos Android e iOS. Além desses motivos, a universalidade de uso do SQLite se dá pelo mesmo ser disponibilizado em domínio público, permitido seu uso gratuito e irrestrito por terceiros (DWAYNE RICHARD HIPP, 2019).

A manipulação de dados espaciais no SQLite pode ser feita através da extensão SpatiaLite (FURIERI, 2015). A mesma possibilita utilizar operadores espaciais nas consultas SQL, tais como distância e intersecção. Essa biblioteca é utilizada pelo SETE na implementação dos algoritmos de roteirização e na edição da malha rural. Vale salientar, que o SpatiaLite é disponibilizado sob a licença de *software* livre *Mozilla Public License 1.1* (OSI, 2019d), que é compatível com a utilizada pelo SETE.

O SQLite serve como uma base de dados local, destinado a salvar os dados no dispositivo do usuário, possibilitando assim tratar os problemas de desempenho e aqueles associados a instabilidade da conexão com a Internet. Entretanto, para realizar o *backup* dos dados é necessário utilizar um sistema na nuvem. Para isso, optou-se pelo Google Firebase (GOOGLE, 2019b; MORONEY, 2017), um banco de dados NoSQL (TIWARI, 2011).

O motivo da adoção dessa tecnologia se dá pela mesma não requerer a contratação de um servidor dedicado para a utilização do serviço da nuvem, uma vez que a Google fornece gratuitamente uma máquina para tal. O Firebase é um *software* livre disponibilizado sob a licença Apache 2.0 (OSI, 2019e), que é compatível com a MIT, a licença utilizada pelo SETE.

A base de dados do Firebase contém um espelho do banco de dados local, descrito pelo SQLite. Periodicamente, o SETE sincroniza os dados salvos localmente com a réplica na nuvem. Através disso, se pode acessar os dados a partir de diferentes sistemas, tais como dispositivos móveis e outros computadores *desktop*. No contexto do SETE, o Firebase serve como uma ponte para recuperar os dados armazenados nos sistemas *desktop*.

Em uma visão geral, observa-se que os atores são descritos com flexibilidade no modelo do *software* SETE. Existem poucos campos obrigatórios, o que possibilita que o sistema possa ser utilizado, mesmo que de forma limitada. A próxima seção descreve as técnicas utilizadas para gerir e criar rotas a partir dos dados cadastrados.

4. Gestão de Rotas

Um dos principais requisitos do *software* de *e-governança* SETE é possibilitar que municípios com diferentes níveis de maturidade e de recursos possam gerir o transporte escolar. Uma parcela significativa dos sistemas existentes é voltada a ação de roteirização, que exige como entrada um georeferenciamento dos atores do sistema (ELLEGOOD *et al.*, 2019; PARK; KIM, 2010) de das rotas.

Nesse sentido, esta seção descreve as ações realizadas pelo SETE com intuito de mitigar esse problema. Especificamente, no *software* SETE, provê quatro níveis para manipular e gerir rotas. O mais simples, prevê apenas o cadastro de dados da mesma, tais como quilometragem e estudantes atendidos. Já o segundo nível, visa possibilitar que o gestor importe os dados da rota a partir de dados GPS registrado por um operador. O terceiro nível permite que o gestor possa desenhar o traçado da rota. Por fim, o último nível realiza a sugestão automática de rotas, isto é, a roteirização utilizando os dados georeferenciados.

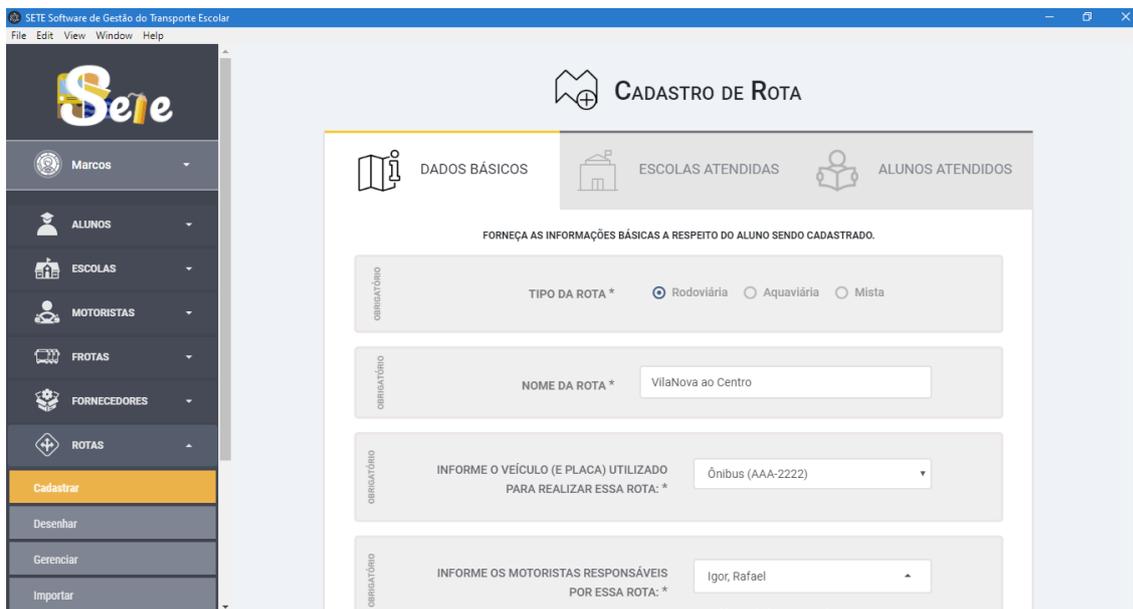
4.1. Cadastro de Rotas

O nível mais simples de gestão requer apenas informações básicas da rota, como descrito no modelo de banco de dados da Seção 3.2.5. Neste nível espera-se do gestor apenas os dados básicos, tais como aqueles referentes aos estudantes atendidos, veículo utilizado, motorista, quilometragem e tempo de percurso da rota.

A ferramenta de cadastro segue o estilo passo a passo (*Wizard*) onde o usuário informa os dados de forma sequencial, conforme ilustrado na Figura 4.1. Em cada passo é feita a validação dos dados, isto possibilita evitar erros de confiabilidade nos mesmos. A validação se dá referentes a obrigatoriedade dos dados e ao seu respectivo formato. No primeiro caso, realiza-se uma verificação se os dados obrigatórios foram informados, já na segunda etapa é feito uma validação do formato dos mesmos. Por exemplo, é feito uma verificação se o dado obrigatório quilometragem da rota se encontra presente e se o mesmo está descrito como um número.

Ao término dessa etapa, o gestor do município terá a rota cadastrada no *software* e poderá dar prosseguimento aos outros níveis de gestão.

Figura 4.1: Tela de cadastro de dados de rota



Fonte: Elaborada pelos autores.

4.2. Importar Rotas

A próxima etapa na gestão das rotas é compreender o traçado das mesmas. Uma maneira de realizar isso é obter o percurso das rotas existentes a partir de dados GPS. Por exemplo, pode-se utilizar os aplicativos móveis *Open GPX Tracker*¹ para iOS e *GPS Tracker*² para Android na obtenção de tais dados.

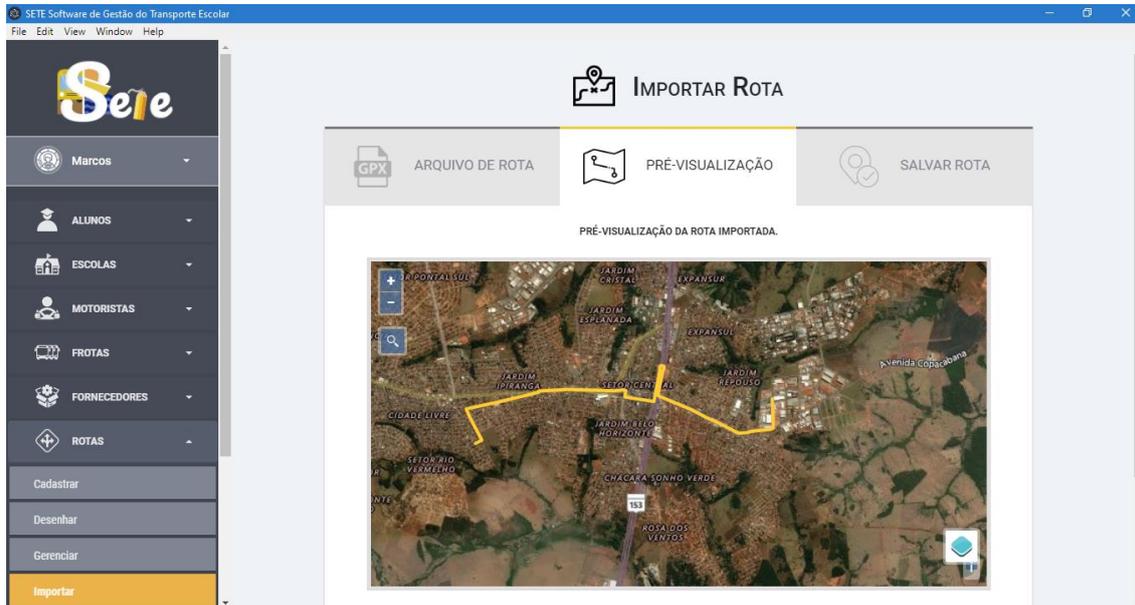
Os dados de percurso da rota devem estar descritos no formato GPS eXchange Format (GPX) (FOSTER, 2004). Este formato é aberto, isto significa que o mesmo pode ser usado por diferentes sistemas sem necessidade de licenciamento. Além disso, destaca-se que o GPX é o formato *de fato* para interoperabilidade e obtenção de percursos de rotas.

O SEITE fornece uma ferramenta para importação desses dados, como ilustrado na Figura 4.2. Primeiramente o sistema lê o dado no formato GPX e verifica se o mesmo não possui erros. Caso positivo, o *software* mostra uma pré-visualização do percurso a ser importado. Em seguida, o gestor escolhe para qual rota, das previamente cadastradas, isto será feito.

¹ Open GPX Tracker – <https://apps.apple.com/br/app/open-gpx-tracker/id984503772>

² GPX Tracker – https://play.google.com/store/apps/details?id=com.gpsserver.gpstracker&hl=pt_BR

Figura 4.2: Tela de importação de rota a partir de dados GPX



Fonte: Elaborada pelos autores.

No término da operação o sistema faz a conversão do percurso em GPX para o formato aberto GeoJSON (BUTLER *et al.*, 2016). Isto é feito pois o formato GeoJSON facilita a utilização de operações espaciais, como obtenção do tamanho e edição da rota. O dado, neste formato é então armazenado no campo SHAPE da rota escolhida.

4.3. Desenho de Rotas

Com intuito de atender aos municípios que não possuem mecanismos para registrar as rotas via GPS, o *software* SETE fornece uma ferramenta gráfica de desenho de rotas. Esta pode ser utilizada não somente para desenhar o percurso da rota, mas para editar aquelas existentes. Isto é importante pois possibilita que os municípios reflitam as mudanças do transporte escolar nas rotas, *e.g.*, editar o percurso para atender um novo estudante.

A ferramenta possibilita o desenho gráfico do percurso com o uso de vários elementos, como ilustrado na Figura 4.3. Os elementos permitem detalhar características da rota, tais como via pavimentada, via não pavimentada, hidrovia, início da rota e impedâncias. Por exemplo, na ilustração parte do percurso da rota está de cor laranja, esta parte significa que este trajeto é realizado sobre uma via rural, enquanto que a parte verde indica que o percurso se dá em área urbana. Já o ícone do burro anuncia a localização da impedância.

Figura 4.3: Tela de desenho e edição de rotas



Fonte: Elaborada pelos autores.

Ao término da edição, o percurso no mapa é convertido para o formato GeoJSON e armazenado na coluna SHAPE da rota escolhida. Além disso, este processo também salva as características das rotas, isto é, o início da rota, tipo de via e impedâncias registradas.

Para implementar tal editor, foi feito o uso da biblioteca de *software* para manipulação de mapas OpenLayers (FARKAS, 2016; METACARTA, 2019). A mesma é disponibilizada sob a licença de *software* livre de duas cláusulas BSD (OSI, 2019b). Para programar o editor foi feita uma extensão das funcionalidades de desenho da biblioteca. Adicionou-se novos elementos de controle no mapa (botões a esquerda) e customizou-se os objetos desenhados no mapa ao clicar nos mesmos. Além disso, adicionou-se a capacidade de mudar o pano de fundo do mapa, de satélite para vetorial e vice-versa.

Apesar da capacidade da ferramenta, alguns municípios, principalmente aqueles com mais recursos, podem desejar explorar mecanismos mais sofisticados para concepção das rotas. Em especial, se os dados dos atores do transporte escolar no município forem georeferenciados, pode-se utilizar a ferramenta de sugestão de rotas.

4.4. Sugestão de Rotas (Roteirização)

A ferramenta de sugestão de rotas possibilita gerar automaticamente um conjunto de rotas baseado nos dados georeferenciados dos atores do transporte escolar no sistema, como estudantes, escolas e motoristas. Somente os municípios que possuem tais dados podem usufruir dessa ferramenta, uma vez que os algoritmos de roteirização requerem estes dados como entrada. Além disso, é necessário prover a malha das vias do município. Para isso, o gestor deve fornecer a mesma no formato do OpenStreetMap ((COAST, 2019).

A tela inicial da ferramenta provê mecanismo para realizar a parametrização do algoritmo de roteirização, como ilustrado na Figura 4.4. Nesta, o gestor pode configurar alguns parâmetros, tais como o conjunto de estudantes alvo (aluno sem rota/todos os alunos), turno dos mesmos (manhã, tarde, noite), o tempo e a distância máxima permitido por rota, a quantidade de veículos e a capacidade máxima de cada veículo.

Ao definir os parâmetros, o gestor pode inicializar o processo de roteirização. Ao fazer isso, o sistema começa a executar o algoritmo de roteirização.

Figura 4.4: Tela de parametrização da roteirização



Fonte: Elaborada pelos autores.

Sabe-se que o problema de roteirização do transporte escolar é um problema NP-difícil (ELLEGOOD *et al.*, 2019; PARK; KIM, 2010). Como consequência, não se conhece um algoritmo que seja capaz de solucioná-lo de forma determinística em uma quantidade de passos polinomial ao tamanho da entrada.

Desta forma, a comunidade investigou mecanismos baseados em meta-heurísticas, como algoritmos genéticos e colônia de formigas para resolvê-los (CHEN *et al.*, 2015; LEWIS; SMITH-MILES; PHILLIPS, 2018; SCHITTEKAT *et al.*, 2013; SIQUEIRA; SOUZA, 2012; SOUZA LIMA *et al.*, 2016). Entretanto, a execução desses algoritmos pode ser problemática no cenário do transporte escolar. Primeiramente, porque requerem um poder computacional, para serem executados, significativamente maior do que aqueles que são tipicamente encontrados nos municípios brasileiros. Segundo, por causa desse poder computacional, mesmo sendo inúmeras vezes mais rápido do que o método combinatório, as heurísticas podem demorar a convergir e não responderem rapidamente ao usuário (SCHITTEKAT *et al.*, 2013; SOUZA LIMA *et al.*, 2016)..

Como alternativa, este trabalho busca implementar um algoritmo exato com intuito de possibilitar que o mesmo seja executado em computadores com diferentes capacidades computacionais. Apesar do resultado ser inferior aos obtidos com métodos baseados em meta-heurísticas, deseja-se produzir um resultado rapidamente ao usuário para evitar que o mesmo tenha de esperar um tempo significativo para a execução do algoritmo.

Desta forma, optou-se por adaptar uma variação do algoritmo SpeedRoute (SHAFABI; WANG; HAGHANI, 2018). As etapas do algoritmo adaptado são descritas na Figura 4.5. Primeiramente, é feito um agrupamento dos atores do transporte escolar em aglomerados. Em seguida, realiza-se a roteirização dos indivíduos de cada grupo. Para tal, decidiu-se

Figura 4.5: Etapas do algoritmo de roteirização



Fonte: Elaborada pelos autores.

utilizar uma variação da heurística de Clarke e Wright voltada ao transporte escolar (BENNETT; GAZIS, 1972; CLARKE; WRIGHT, 1964). Por fim, realiza-se uma otimização individual das rotas geradas utilizando algoritmo de busca local (CROES, 1958). Para compreender o funcionamento do algoritmo, as subseções seguintes discorrem concisamente do papel de cada parte.

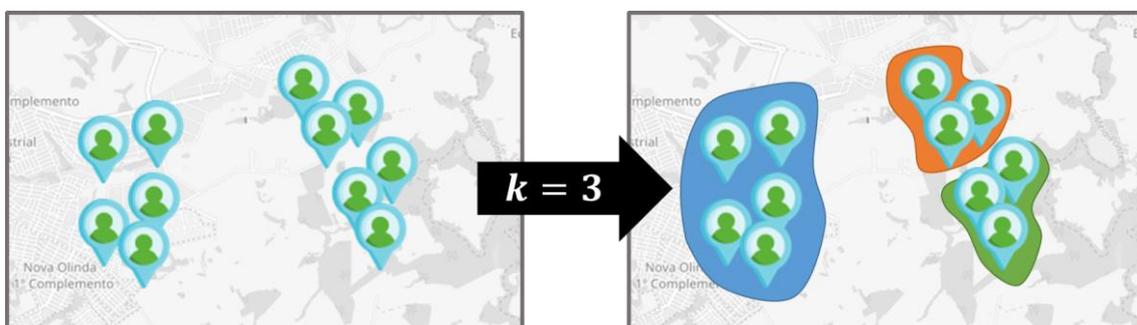
4.4.1. Aglomeração (*k-Means*)

A primeira etapa do algoritmo visa dividir a população de estudantes do transporte escolar em diferentes *clusters* (aglomerados). Esse processo é feito antes de realizar a roteirização em si com intuito de acelerar o cálculo que cálculo a combinação de rotas, uma vez que diminui significativamente o número de variáveis.

O algoritmo de aglomeração adotado no trabalho é o *k-Means* (JAIN, 2010), que divide o universo de dados, neste caso os n estudantes, em k conjuntos. Por exemplo, a Figura 4.6 ilustra um cenário com 11 estudantes que são divididos em 3 *clusters*. Em termos gerais, o algoritmo visa associar os n estudantes do problema a k conjuntos C_1, C_2, \dots, C_k de maneira que um dado estudante p está em um cluster C_i se e somente se a distância de p até a centroide z_i de C_i é menor que a distância até as centroides z_j de todos os *clusters* C_j restantes, para todo $j \neq i$. Isto pode ser expresso no seguinte problema de minimização:

$$\arg \min_C \sum_{i=1}^k \sum_{p \in C_i} \|p - z_i\|^2 \quad (1)$$

Figura 4.6: Exemplo de execução do algoritmo de aglomeração *k-Means*



Fonte: Elaborada pelos autores.

No problema do transporte escolar, associamos a quantidade de aglomerados k ao número de veículos. Já os pontos p_1, p_2, \dots, p_n a serem agrupados representam os n estudantes. Computa-se a função de distância de um ponto p a um ponto q na malha m através da função $dist(p, q, m)$.

O algoritmo para solucionar esse problema segue os passos ilustrados na Figura 4.7. Primeiramente, seleciona-se k estudantes de forma aleatória e utiliza as posições do mesmo para gerar os clusters e as centroides z_1, z_2, \dots, z_k iniciais. Em seguida, inicializa-se o custo da função objetivo S como infinito. Realiza-se a distribuição dos estudantes aos clusters correspondentes usando a função de distância. Posteriormente computa-se as novas centroides e o novo custo S' . Caso o custo seja inferior, o algoritmo converge e retorna os aglomerados C_1, C_2, \dots, C_k , caso contrário verifica-se novamente a pertinência dos estudantes aos novos *clusters*.

Figura 4.7: Algoritmo *k-Means*.

Entrada:	k veículos, p estudantes, m malha
Saída:	C_1, C_2, \dots, C_k conjuntos de estudantes
1	Selecione k estudantes z_1, z_2, \dots, z_k como centroides iniciais e crie os clusters correspondentes C_1, C_2, \dots, C_k
2	Atribua um custo inicial infinito $S \leftarrow \infty$
3	Atribua o ponto p ao cluster C_i se e somente se $dist(p, z_i, m) < dist(p, z_j, m) \forall z_j \mid j = 1, \dots, k$ e $j \neq i$
4	Compute as novas centroides z'_1, z'_2, \dots, z'_k como: $z'_i = \frac{1}{ C_i } \times \sum_{p \in C_i} p$
5	Compute o novo custo $S' = \sum_{i=1}^k \sum_{p \in C_i} \ p - z_i\ ^2$
6	Se $S' = S$ termine (ocorreu convergência) e retorne C_1, C_2, \dots, C_k
7	Senão $S \leftarrow S'$ e volte para o passo 3.

Fonte: Adaptado de (HAN; KAMBER; PEI, 2011; JAIN, 2010).

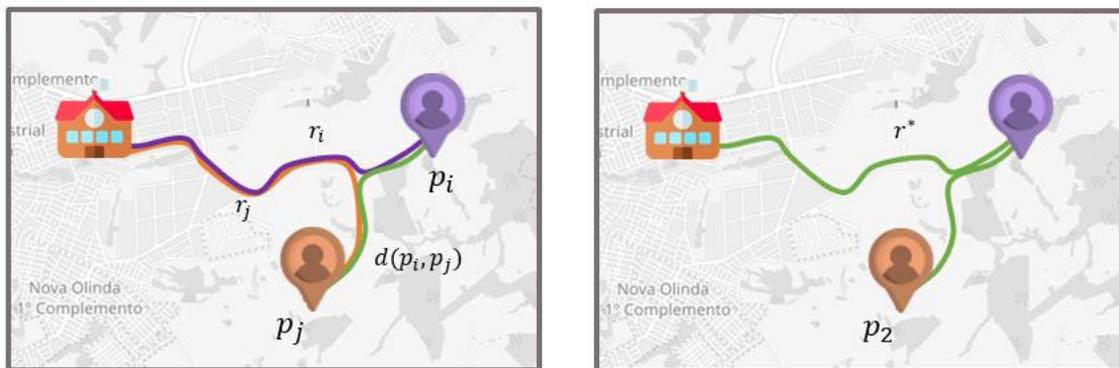
4.4.2. Clarke e Wright

Em posse dos conjuntos C_1, C_2, \dots, C_k de estudantes, a próxima etapa consiste em executar a roteirização em cada um dos aglomerados. Como entrada dessa etapa, espera-se a malha viária m , a posição de embarque dos estudantes $p \in C$ do *cluster* em análise, a posição das escolas e_1, e_2, \dots, e_j dos alunos presente no conjunto e a garagem g dos veículos.

Para realizar a roteirização, optou-se por utilizar uma variação da heurística de Clarke e Wright para o problema do transporte escolar (BENNETT; GAZIS, 1972). O motivo para tal é a rapidez e simplicidade do algoritmo. A heurística de Clarke e Wright (CLARKE; WRIGHT, 1964) inicia o processo de roteirização gerando uma rota individual para cada estudante até a respectiva escola do mesmo. Em seguida, calcula-se a economia $s(i, j)$ obtida ao mesclar as rotas do estudante i com o j .

Por exemplo, se a distância da escola ao ponto de embarque do estudante i e ao ponto de embarque de j é respectivamente 20 km e 25 km e as distâncias diretas entre os pontos i e j é de 7 km a economia em termos de quilometragem de juntar as duas rotas é de $20 + 25 - 7 = 38$ km. A Figura 4.8 ilustra esse cenário. Observe que se o algoritmo combinar essas duas rotas terá uma redução (economia) de 38 km, uma vez que não será necessário ter rotas individuais que atendam os pontos de embarque p_i e p_j .

Figura 4.8: Cenário exemplo do algoritmo Clarke e Wright



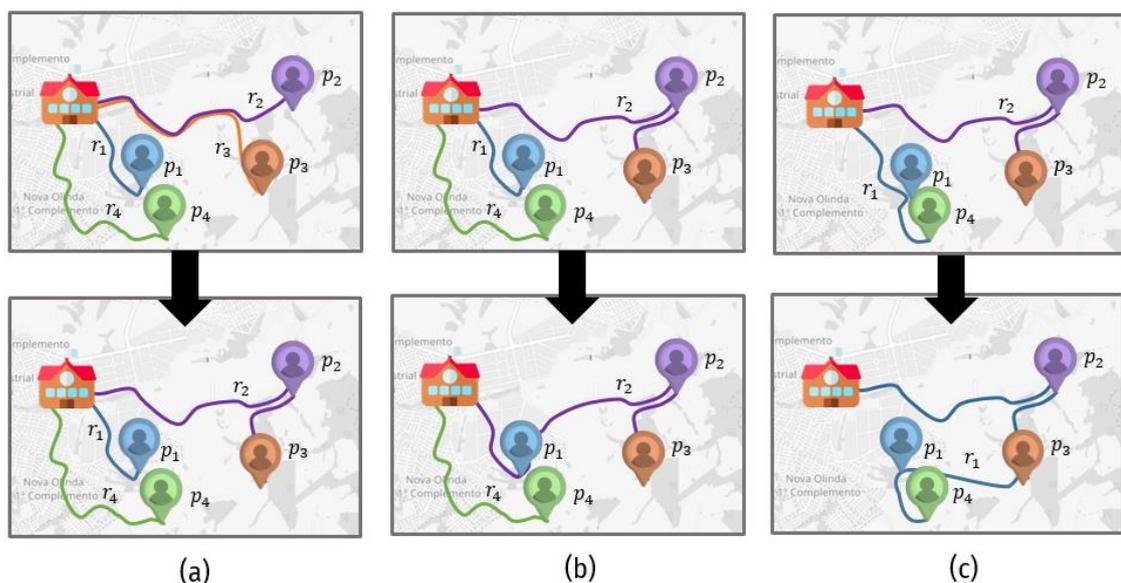
Fonte: Elaborada pelos autores.

O algoritmo realiza os seguintes passos (LARSON; ODONI, 1981). Primeiramente, calcula-se a economia $s(i, j) = dist(e, i, m) + dist(e, j, m) - dist(i, j, m)$ para todos os pares, onde $dist(i, j, m)$ representa a distância entre os pontos p_i e p_j na malha m e a variável e representa a localização da escola. Em seguida, ordena-se a lista de redução de maneira decrescente, isto é, os pares que reduzem maior a distância percorrida são analisados primeiro. Após isso inicie o processamento dos itens da lista. O algoritmo realizará a combinação das rotas indicadas pelo par $s(i, j)$ se e somente se nenhuma das restrições de capacidade foram violadas (e.g., tempo e distância máxima) e:

- (a) O ponto de embarque de i e o de j não foram combinados a nenhuma rota. Neste caso se cria a rota que mescla os dois locais; ou
- (b) Somente um dos estudantes i ou j foi mesclado a uma rota e a posição do ponto do mesmo dentro desta se encontra na sua extremidade, ou seja, é o primeiro ou último ponto de embarque antes da escola. Neste caso, cria-se o arco (i, j) na rota.
- (c) Os dois pontos de embarque já foram mesclados com outros estudantes, isto é, encontram-se em outras rotas. Entretanto, isto é feito somente se um dos pontos for o primeiro ponto de embarque e outro o último, desta forma, é possível ligar as extremidades da rota sem quebrá-la (último para o primeiro ponto))

Os três cenários de combinação de rota são ilustrados na Figura 4.9. No item (a) observa-se a possibilidade de juntar as rotas individuais r_2 e r_3 , uma vez que as mesmas ainda não foram mescladas. No item (b), ao processar o par $s(p_1, p_2)$ pode-se mesclar as rotas de $r_1 = \{p_1\}$ com a de $r_2 = \{p_2, p_3\}$, uma vez que apesar do ponto p_2 pertencer a uma rota combinada, o mesmo se encontra na extremidade desta. No item (c), suponha a existência prévia das rotas $r_1 = \{p_1, p_4\}$ e $r_2 = \{p_2, p_3\}$, ao processar o par $s(p_3, p_1)$ pode-se mesclar as rotas dos dois pontos de embarque, uma vez que p_3 e p_1 são o primeiro e último ponto de embarque respectivamente de suas rotas. Observe que a combinação só será feita se não violar as capacidades do veículo e distância e tempo máximo informado.

Figura 4.9: Três cenários de combinação de rota do algoritmo Clarke e Wright



Fonte: Elaborada pelos autores.

Observa-se que a ideia geral do algoritmo é combinar as rotas individuais geradas de modo que elas apresentem a maior economia. Por isso, o algoritmo é classificado como guloso, uma vez que tenta tomar a melhor decisão com intuito de minimizar a quilometragem ou o tempo global utilizado.

Tradicionalmente, o algoritmo de Clarke e Wright foi projetado para a roteirização de produtos, onde se tinha um depósito e os veículos retornavam para a mesma (CLARKE; WRIGHT, 1964). A variação proposta para atender o transporte escolar (BENNETT; GAZIS, 1972) realiza as seguintes modificações ao algoritmo:

- O algoritmo inicia a construção da rota a partir da garagem;
- Ao invés de retornar para a garagem, o último arco da rota direciona-se a escola;
- Possibilidade de tratar de múltiplas escolas na rota ao representar as mesmas com um super-destino (centroide);

Além disso, esta variação do algoritmo adiciona uma análise do ganho de inversão da rota em processamento. Isto é importante pois permite tratar situações em que não é possível realizar a combinação de rotas. Imagine o cenário descrito na Figura 4.10. Nele, parece ser desejável mesclar as rotas r_1 e r_2 ao analisar o par $s(p_2, p_3)$. Entretanto, como ambos são os primeiros pontos de embarque da rota, não é possível fazer tal combinação. Logo, para possibilitar isso, verifica-se a possibilidade de inverter as rotas. Neste caso, ao inverter a rota $r_2^* = \{p_4, p_3\}$ pode-se fazer a combinação das duas.

Figura 4.10: Exemplo de cenário onde a inversão de rota é vantajosa



Fonte: Adaptado de (BENNETT; GAZIS, 1972).

Como resultado, o algoritmo produz uma rota com a ordem dos pontos de parada dos respectivos estudantes e escolas. Esta rota pode ser ainda melhorada através de algoritmos de pós-processamento, detalhada a seguir.

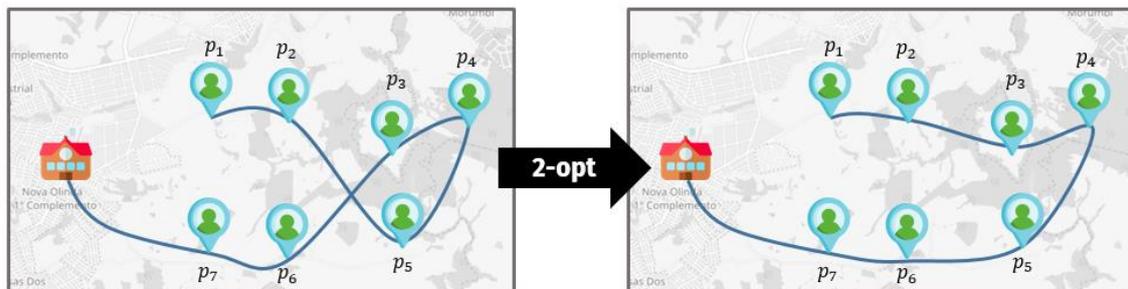
4.4.3. Busca Local (2-opt)

Após a construção da rota inicia-se o processo de pós-processamento. Aqui, optou-se pelo algoritmo de busca local 2-opt para esta tarefa (CROES, 1958). Este algoritmo clássico tem como objetivo inicial remover arestas que se cruzam na rota. Para realizar isso, o método realiza a troca de duas arestas (daí o número 2) com outras duas criando assim uma rota alternativa. Caso o custo desta rota seja considerado menor do que a original, a troca é efetuada. Caso negativo, a rota original é mantida.

Por exemplo, considere o cenário ilustrado na Figura 4.11. Nela, a aresta (p_2, p_5) cruza a aresta (p_3, p_6) . Neste caso pode ser vantajoso para o algoritmo retirar esse cruzamento.

Para realizar isso automaticamente, o algoritmo utiliza a seguinte lógica. Primeiramente, realiza-se a combinação de todos os pares de posição (índices) da rota. Tais pares indicam os pontos de paradas (i, j) que serão analisados. Por exemplo, para ilustração considere que estamos analisando o par $(3, 5)$ da rota $r = \{p_1, p_2, p_5, p_4, p_3, p_6, p_7\}$. Neste caso, tal par indica os pontos de parada (p_5, p_3) . Então o algoritmo divide a rota em três subrotas: $[r_1, r_2, r_3] = [(1 \dots i - 1), (i, j)^R, (j + 1 \dots n)]$. A primeira subrota contém os elementos do índice 1 até $i - 1$, neste caso de $(1, 2)$, a saber $r_1 = \{p_1, p_2\}$. Em seguida inverte-se a subrota $r_2 = \{p_5, p_4, p_3\}$ de i até j , produzindo assim $r_2^R = \{p_3, p_4, p_5\}$. Por fim, r_3 representa a última parte da rota, que é composta pelos elementos restantes. Neste caso,

Figura 4.11: Exemplo de uso do algoritmo 2-opt



Fonte: Adaptado de <<http://tsp-basics.blogspot.com/2017/03/2-opt-move.html>>.

$r_3 = \{p_6, p_7\}$. A rota alternativa é a combinação das três: $r_a = \{p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$. Caso a rota alternativa possua um custo menor que a rota original se realiza a troca de posição dos pontos de parada.

Observe que as subrotas r_1 e r_3 não são modificadas na operação. O que ocorre é um corte na rota original e a inversão da subrota do meio r_2 , produzindo assim a rota alternativa $r_a = [r_1, r_2^R, r_3]$, onde r_2^R indica o inverso da subrota r_2 . Essas operações são custosas, uma vez que é feito uma comparação par a par dos índices da rota. Entretanto, como a mesma é restrita aos pares de uma única rota, o número de variáveis é pequeno. Isto porque, tipicamente, a quantidade de pontos de embarque dos estudantes é inferior a capacidade do veículo. Como o tamanho das rotas, em termo de pontos de embarque, é proporcional a capacidade do veículo, se tem uma quantidade pequena de pares para comparação.

Como saída, a busca local do algoritmo 2-opt produzirá uma versão otimizada da rota obtida pela etapa anterior. Esta rota é apresentada para o gestor na tela rota geradas da ferramenta sugestão, conforme descrito na seção seguinte.

4.4.4. Resultado da Roteirização

Após o término da roteirização, o *software* SETE apresenta as rotas geradas na ferramenta para o usuário, conforme ilustrado na Figura 4.12. Os percursos das rotas geradas podem ser salvos utilizando a ferramenta gráfica. É importante destacar que não é necessário aceitar todas as rotas sugeridas, isto é, o gestor pode escolher apenas uma ou outra rota gerada.

Além disso, a ferramenta possibilita que o gestor possa acompanhar e entender passo a passo o deslocamento dos veículos na rota. Por exemplo, o usuário pode clicar no botão acompanhar rota para inicializar uma animação de um veículo, no caso da ilustração de um ônibus, pelo percurso da rota.

Por fim, o gestor pode clicar na rota sugerida para obter informações a respeito da mesma, como distância, número de alunos atendidos e quilometragem morta, isto é, quanto que o veículo tem de percorrer até chegar ao primeiro estudante da rota.

Figura 4.12: Tela de resultado de sugestão de rotas



Fonte: Elaborada pelos autores

A próxima seção traz uma discussão a respeito de outros dados e informações estatísticas que o gestor pode utilizar para auxiliar na gestão e na tomada de decisão.

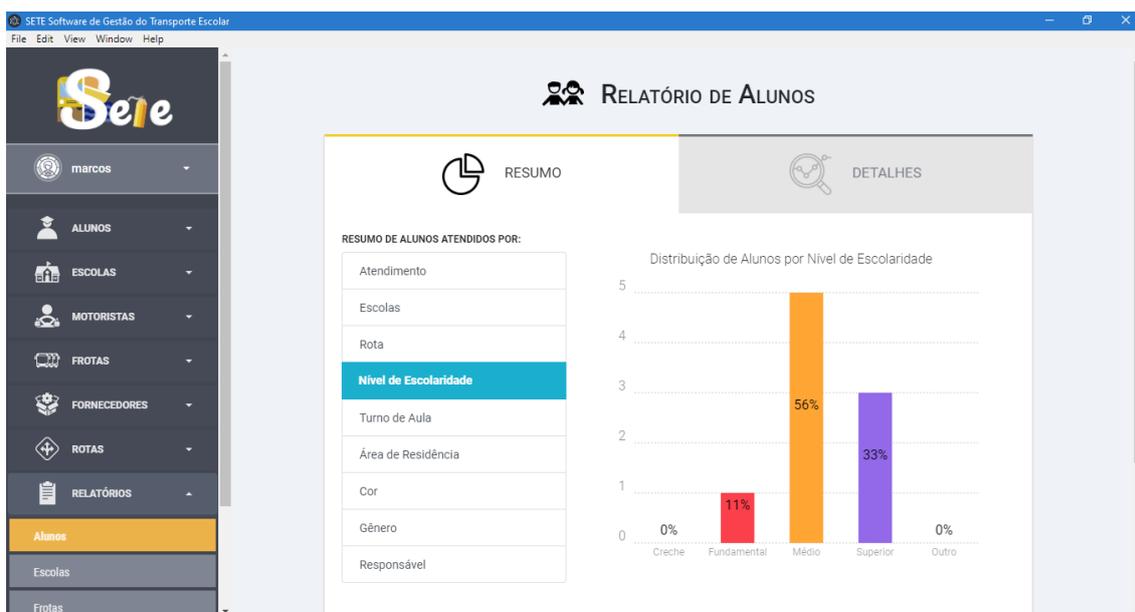
5. Relatórios

Os relatórios são elementos primordiais em sistemas de *e-governança*, isto porque visam sintetizar as informações para os gestores. Informações como número médio de alunos atendidos por rota, quilometragem de cada viagem, horário e detalhamento de cada linha (rural ou urbana) são essenciais para uma boa e adequada tomada de decisão. Nesse sentido, o *software* SETE visa apresentar relatórios com indicadores e gráficos dos atores do transporte escolar.

Cada ator do transporte escolar possui uma aba de relatório no sistema. Esta aba é dividida em duas partes, resumo e detalhes, conforme ilustrado na Figura 5.1. O primeiro item visa sintetizar algumas características do ator. Por exemplo, a figura apresenta um gráfico de barras relatando a distribuição porcentual dos alunos atendidos por nível de escolaridade, *e.g.*, fundamental, médio e superior.

Já a aba detalhes possibilita que o gestor possa investigar cada estatística. Por exemplo, através desta ferramenta o mesmo pode identificar quais são os estudantes do período noturno que não são atendidos. É válido destacar que todos os detalhes dos relatórios gerados pelo *software* podem ser exportados para planilhas eletrônicas e arquivos PDF.

Figura 5.1: Tela de relatórios de alunos



Fonte: Elaborada pelos autores

As seções subsequentes destacam os indicadores e estatísticas produzidas pelo sistema para cada ator.

5.1. Estudantes

Na gestão do transporte escolar, os estudantes são os principais atores do sistema. Nesse sentido, é importante para o gestor conhecer informações a respeito do atendimento e da qualidade do serviço (como lotação). Assim, o sistema apresenta as seguintes estatísticas:

- **Alunos atendidos:** este indicador identifica a clientela atendida pelo transporte escolar, assim é feita uma listagem dos alunos atendidos e suas características, proporcionando um detalhamento e porcentagem dos mesmos. Define-se como atendimento a presença do aluno a uma rota e escola no sistema.
- **Número médio de alunos por escola:** apresenta o número médio de estudantes transportados por cada escola cadastrada no sistema.
- **Número médio de alunos por rota:** apresenta a quantidade média de estudantes transportados por rota.
- **Nível de escolaridade:** aqui apresenta-se a distribuição dos alunos transportados por nível de escolaridade (creche, fundamental, médio, superior e outros).
- **Turno de estudo:** este indicador apresenta a porcentagem e detalhe dos alunos atendidos por turno de ensino (matutino, vespertino, integral e noturno).
- **Área de residência:** este valor apresenta a porcentagem e distribuição de alunos por local de residência (área urbana e rural).
- **Cor:** detalhamento e porcentagem de alunos atendidos por cor da pele.
- **Gênero:** detalhamento porcentagem de alunos atendidos por gênero.
- **Responsável:** apresenta a distribuição dos responsáveis (pai, mãe, tia, avó, primo, etc) dos estudantes cadastrados.

5.2. Escolas

Em relação as escolas, o sistema apresenta os seguintes indicadores e gráficos:

- **Localidade das escolas:** este indicador apresenta o detalhamento e a porcentagem de escolas atendidas pelo transporte escolar por região (área urbana e área rural).
- **Dependência das escolas:** este valor apresenta a porcentagem e o detalhamento das escolas atendidas por tipo de dependência, isto é, se as escolas são federais, estaduais, municipais ou privadas.
- **Nível de ensino:** indica a distribuição de escolas por nível de ensino ofertado, como ensino fundamental, médio e superior.
- **Regime de ensino:** apresenta a porcentagem de escolas atendidas pelo transporte por regime de ensino, como regular, EJA e ensino profissionalizante.
- **Horário de funcionamento:** destaca a distribuição de escolas atendidas pelo transporte por horário, manhã, tarde e noite.

5.3. Motoristas

A aba de motoristas apresenta as seguintes estatísticas:

- **Distribuição de Idade:** este indicador apresenta a idade média, mínima e máxima dos condutores dos veículos.
- **Distribuição de habilitação:** indica a porcentagem de habilitação de CNH por categoria obtidas pelos condutores do município.
- **Quantidade por rota:** indica a distribuição dos motoristas por rota, se existe ou não a presença de vários condutores por trajeto.
- **Quantidade por turno:** apresenta o detalhamento e a porcentagem de motoristas que trabalham nos turnos da manhã, tarde e noite.

5.4. Veículos

A composição da frota é um elemento chave na gestão. Desta forma, o sistema SETE apresenta as seguintes estatísticas para o gestor:

- **Capacidade atual:** este valor indica a quantidade e a média de estudantes transportados ao considerar os veículos da frota.
- **Capacidade média:** este indicador apresenta a média da capacidade máxima de assentos dos veículos. Por exemplo, o valor 37.5 indica que em média cada veículo fornece 37.5 assentos.
- **Categoria da frota:** apresenta a distribuição de veículos por categoria (ônibus, micro-ônibus, lancha, barco, *etc*).
- **Idade:** este indicador apresenta a idade média da frota cadastrada.
- **Marca dos veículos:** indica a distribuição de veículos da frota por marca, como Iveco, Mercedes-Benz, Renault, *etc*.
- **Modelos utilizados:** apresenta a porcentagem de veículos do programa Caminho da Escola presentes na frota, em função dos modelos do programa, como ORE 1, ORE 2, Lancha a Gasolina, *etc*.
- **Origem do veículo:** indica a quantidade e porcentagem de veículos classificados em frota própria e em frota terceirizada.
- **Manutenção média:** apresenta

5.5. Rotas

As rotas são elementos cruciais na operação do transporte escolar. Desta maneira, decidiu-se focar em questões de tempo e percurso da mesma para auxiliar na tomada de decisão dos gestores. O *software* apresenta os seguintes indicadores e gráficos para as rotas:

- **Quilometragem percorrida:** indica a quilometragem total e média das rotas cadastradas no sistema.
- **Tempo gasto:** apresenta como indicador o tempo total e médio em minutos das rotas cadastradas.
- **Dificuldade de rotas:** apresenta a porcentagem de dificuldade de acessos registradas pelas rotas do município.
- **Trecho morto:** ilustra a quilometragem total e média do trecho morto das rotas, isto é, a distância percorrida pelos veículos da garagem até o primeiro ponto de embarque dos estudantes.

5.6. Garagem

As garagens são atores simples no *software*. O impacto maior das mesmas na gestão situa na localização e distância de início das rotas. Desta forma, o *software* SETE apresenta as seguintes estatísticas:

- **Distância média as rotas:** este indicador apresenta a distância média das garagens as rotas atendidas pelos veículos da mesma.
- **Quantidade de veículos:** apresenta como valor a quantidade de veículos atendidos por garagem cadastrada no sistema.

5.7. Fornecedor

No *software* SETE os fornecedores atuam especificamente com os veículos. Em relação as escolas, o sistema apresenta os seguintes indicadores e gráficos:

- **Tipo de fornecer:** apresenta uma distribuição porcentual das áreas de atuação dos fornecedores cadastrados, tais como mecânica, seguros e combustível.
- **Quantidade de serviços** esse indicador apresenta a porcentagem e o número de serviços realizados pelos fornecedores.

6. Considerações Finais

O relatório apresentou o projeto de engenharia de software do *software* de *e-governança* SETE. Espera-se que o sistema possa ser utilizado por municípios com diferentes níveis de maturidade de gestão do transporte escolar, desde os mais simples com pouquíssimos dados até os mais avançados que possuem mecanismos para georeferenciar os estudantes, escolas e veículos.

A arquitetura do *software* visou ser flexível o bastante para possibilitar o uso frente as dificuldades dos municípios, como falta de dados e conexão instável com a Internet. A utilização do banco de dados local possibilita que o sistema possa operar mesmo com a ausência de conectividade.

Com intuito de evitar retrabalho, o sistema foi projetado com mecanismos de importação e exportação de dados. Especificamente, tentou-se reutilizar as variáveis existentes em fonte de dados oficiais, como as do CENSO escolar. Na outra direção, todos os dados cadastrados no sistema podem ser exportados em planilhas eletrônicas e em documentos PDF.

Por fim, conclui-se o objetivo de não depender de *software* proprietário para utilizar o sistema. Isto possibilita que o mesmo possa ser implantado em municípios sem custo e sem necessidade de licenciar *softwares* adicionais. O SETE é distribuído sob a licença de *software* livre MIT (OSI, 2019a), que possibilita não somente a redistribuição gratuita, mas o compartilhamento e modificação do código fonte do parte dos municípios.

Como próxima etapa, pretende-se expandir as funcionalidades do SETE nos dispositivos móveis, *e.g.*, permitindo realizar operações de importação de rota diretamente dos dados GPS do aparelho. Também será feito um estudo piloto com municípios locais para estudar as dificuldades de implantação do *software* baseado na realidade dos municípios.

Referências Bibliográficas

ALLEN, G.; OWENS, M. **The Definitive Guide to SQLite**. 2nd. ed. Berkely, CA, USA: Apress, 2010.

AMÉRICO, J. **O que está sendo feito para melhorar a internet brasileira?** Disponível em: <<https://olhardigital.com.br/pro/noticia/o-que-esta-sendo-feito-para-melhorar-a-internet-brasileira/58387>>. Acesso em: 15 out. 2019.

BATINI, C.; CERI, S.; NAVATHE, S. B. **Conceptual Database Design: An Entity-relationship Approach**. Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc., 1992.

BENNETT, B. T.; GAZIS, D. C. School bus routing by computer. **Transportation Research**, v. 6, n. 4, p. 317–325, 1972.

BRASIL (1988). Constituição Federal do Brasil, 1988.

BUTLER, H. et al. **The GeoJSON Format**. [s.l.: s.n.]. Disponível em: <<https://tools.ietf.org/rfc/rfc7946.txt>>. Acesso em: 20 nov. 2019.

CARVALHO, W. L.; LEITE, P. DE S.; NASCIMENTO, H. P. **EVOLUÇÃO DO TRANSPORTE ESCOLAR RURAL BRASILEIRO NO MODO RODOVIÁRIO**. Anais do XXIX Congresso Nacional de Pesquisa em Transporte da Anpet. **Anais...Ouro Preto: ANPET**, 2015

CERVANTES, H.; KAZMAN, R. **Designing Software Architectures: A Practical Approach**. 1st. ed. [s.l.] Addison-Wesley Professional, 2016.

CGI.BR. **TIC GOVERNO ELETRÔNICO - Pesquisa sobre o uso das tecnologias de informação e comunicação no setor público brasileiro**. [s.l.] Comitê Gestor da Internet no Brasil, 2018.

CHEN, P. P.-S. The Entity-relationship Model - Toward a Unified View of Data. **ACM Trans. Database Syst.**, v. 1, n. 1, p. 9–36, 1976.

CHEN, X. et al. Exact and metaheuristic approaches for a bi-objective school bus scheduling problem. **PLoS ONE**, v. 10, n. 7, p. 1–20, 2015.

CLARKE, G.; WRIGHT, J. W. Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. **Operations Research**, v. 12, n. 4, p. 568–581, 1964.

COAST, S. **OpenStreetMap**. Disponível em: <<https://www.openstreetmap.org/>>. Acesso em: 20 nov. 2019.

CROES, G. A. A Method for Solving Traveling-Salesman Problems. **Operations Research**, v. 6, n. 6, p. 791–812, 1958.

CUNHA, M. A.; QUANDT, C. O. **O Perfil dos Municípios Brasileiros em Relação ao Uso e à Infraestrutura de TIC: uma Análise dos Clusters**. Anais do XXXVIII Encontro da ANPAD. **Anais...Rio de Janeiro, Brasil: 2014**

DATE, C. J. **An Introduction to Database Systems**. 8. ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.

DWAYNE RICHARD HIPPI. **SQLite**. Disponível em: <<https://www.sqlite.org/>>. Acesso em: 20 set. 2019.

ELLEGOOD, W. A. et al. School bus routing problem: Contemporary trends and research directions.

Omega (United Kingdom), 2019.

ESRI. **ArcGIS Desktop Subscription**. Disponível em: <<https://www.esri.com/en-us/arcgis/products/arcgis-desktop-subscription>>. Acesso em: 4 fev. 2019.

EVEREST, G. **BASIC DATA STRUCTURE MODELS EXPLAINED WITH A COMMON EXAMPLE**. Fifth Texas Conference on Computing Systems. **Anais...Texas**, USA: IEEE Computer Society Publications Office, 1976

FARKAS, G. **Mastering OpenLayers 3**. [s.l.] Packt Publishing, 2016.

FÁTIMA, V. R. **Análise de Requisitos para o desenvolvimento para avaliação e Gestão Escolar**. [s.l.] Universidade Federal da Fronteira Sul, 2015.

FERREIRA, C. O. **TRACK2SCHOOL: SISTEMA DE CONTROLE PARA TRANSPORTE ESCOLAR**. [s.l.] Universidade Federal do Paraná, 2016.

FNDE/CEFTRU. *Projeto: Transporte Escolar Rural – Volume II – Questionário Web*. Centro de Formação de Recursos Humanos e Fundo Nacional de Desenvolvimento da Educação, 2007a.

FNDE/CEFTRU. *Projeto: Transporte Escolar Rural – Volume III – Tomo I – Caracterização do Transporte Escolar nos Municípios Visitados*. Centro de Formação de Recursos Humanos e Fundo Nacional de Desenvolvimento da Educação, 2007b.

FNDE/CEFTRU. *Diagnóstico do Transporte Escolar Rural – Volume I*. Centro de Formação de Recursos Humanos e Fundo Nacional de Desenvolvimento da Educação, 2008.

FOSTER, D. **GPX: the GPS Exchange Format**. [s.l.: s.n.]. Disponível em: <<https://www.topografix.com/gpx.asp>>.

FURIERI, A. **SpatiaLite**. Disponível em: <<https://www.gaia-gis.it/fossil/libspatialite/index>>. Acesso em: 20 nov. 2019.

GHEZZI, C.; JAZAYERI, M.; MANDRIOLI, D. **Fundamentals of Software Engineering**. 2nd. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2002.

GITHUB. **Electron**. Disponível em: <<https://electronjs.org/>>. Acesso em: 20 nov. 2019.

GOOGLE. **Google Maps Plataforma**. Disponível em: <<https://cloud.google.com/maps-platform/pricing/sheet/?hl=pt-br>>. Acesso em: 5 fev. 2019a.

GOOGLE. **Firebase**. Disponível em: <<https://firebase.google.com/>>. Acesso em: 20 nov. 2019b.

GRASER, A.; STRAUB, M.; DRAGASCHNIG, M. Is OSM Good Enough for Vehicle Routing? A Study Comparing Street Networks in Vienna. In: GARTNER, G.; HUANG, H. (Eds.). . **Progress in Location-Based Services 2014**. Cham: Springer International Publishing, 2015. p. 3–17.

HAKLAY, M.; WEBER, P. OpenStreetMap: User-Generated Street Maps. **IEEE Pervasive Computing**, v. 7, n. 4, p. 12–18, 2008.

HAN, J.; KAMBER, M.; PEI, J. **Data Mining: Concepts and Techniques**. 3rd. ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2011.

INEP. **Censo Escolar Resultado e Resumos**. Disponível em: <<http://portal.inep.gov.br/censo-escolar>>. Acesso em: 7 jan. 2019.

JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651–666, jun. 2010.

JAMBEIRO, O.; SOBREIRA, R.; RABELO, P. TICs e Gestão Pública em Planos Diretores de Cidades Brasileiras. **Revista de Economía Política de las Tecnologías de la Información y Comunicación**, v. XI, n. n.3, 2009.

JENSEN, P. B. **Cross-Platform Desktop Applications: Using Node, Electron, and NW.js**. 1st. ed. Greenwich, CT, USA: Manning Publications Co., 2017.

KRALIDIS, A. T. Geospatial Open Source and Open Standards Convergences. In: HALL, G. B.; LEAHY, M. G. (Eds.). . **Open Source Approaches in Spatial Data Handling**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 1–20.

LARSON, R. C.; ODONI, A. R. **Urban operations research**. [s.l.] Prentice Hall PTR, 1981.

LEWIS, R.; SMITH-MILES, K.; PHILLIPS, K. The school bus routing problem: An analysis and algorithm. **Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)**, v. 10765 LNCS, n. 1, p. 287–298, 2018.

METACARTA. **OpenLayers**.

MOREIRA, A. N. H.; JÁCOMO, S. DE A.; GRIEBELER, N. P. SISTEMA DE INFORMAÇÃO GEOGRÁFICA PARA GESTÃO DE ROTAS DO TRANSPORTE ESCOLAR RURAL. **Revista Brasileira de Cartografia**, v. 68, n. 2, p. 421–435, 2016.

MORONEY, L. **The Definitive Guide to Firebase: Build Android Apps on Google’s Mobile Platform**. 1st. ed. Berkely, CA, USA: Apress, 2017.

NEIS, P.; ZIELSTRA, D. Recent Developments and Future Trends in Volunteered Geographic Information Research: The Case of OpenStreetMap. **Future Internet**, v. 6, n. 1, p. 76–106, 2014.

OPEN KNOWLEDGE FOUNDATION. **Open Data Commons Open Database License (ODbL)**. Disponível em: <<https://opendatacommons.org/licenses/odbl/index.html>>. Acesso em: 10 mar. 2019.

OSI. **The MIT License**. Disponível em: <<https://opensource.org/licenses/MIT>>. Acesso em: 10 mar. 2019a.

OSI. **The 2-Clause BSD License**. Disponível em: <<https://opensource.org/licenses/BSD-2-Clause>>. Acesso em: 5 set. 2019b.

OSI. **GNU General Public License version 2**. Disponível em: <<https://opensource.org/licenses/gpl-2.0.php>>. Acesso em: 5 set. 2019c.

OSI. **Mozilla Public License 1.1**. Disponível em: <<https://opensource.org/licenses/MPL-1.1>>. Acesso em: 10 mar. 2019d.

OSI. **Apache License, Version 2.0**. Disponível em: <<https://opensource.org/licenses/Apache-2.0>>. Acesso em: 10 mar. 2019e.

PARK, J.; KIM, B. I. The school bus routing problem: A review. **European Journal of Operational Research**, v. 202, n. 2, p. 311–319, 2010.

PORTO, M. F. et al. Developing a GIS for rural school transportation in Minas Gerais, Brazil. **Journal of Systemics, Cybernetics and Informatics**, v. 13, n. 2, p. 89–94, 2015.

PRZEYBILOVICZ, E.; CUNHA, M. A.; MEIRELLES, F. DE S. O uso da tecnologia da informação e comunicação para caracterizar os municípios: quem são e o que precisam para desenvolver ações de governo eletrônico e smart city. **Revista de Administração Pública**, v. 52, n. 4, p. 630–649, ago. 2018.

SCHITTEKAT, P. et al. A metaheuristic for the school bus routing problem with bus stop selection. **European Journal of Operational Research**, v. 229, n. 2, p. 518–528, 2013.

SHAFABI, A.; WANG, Z.; HAGHANI, A. SpeedRoute: Fast, efficient solutions for school bus routing problems. **Transportation Research Part B: Methodological**, v. 117, p. 473–493, 2018.

SILBERSCHATZ, A.; KORTH, H.; SUDARSHAN, S. **Database Systems Concepts**. 5. ed. New York, NY, USA: McGraw-Hill, Inc., 2006.

SIQUEIRA, P. H.; SOUZA, L. V. DE. Métodos Heurísticos aplicados na construção de roteiros de Transporte Escolar para o Estado do Paraná. **Transportes**, v. 20, n. 3, p. 28–40, 2012.

SOMMERVILLE, I.; SAWYER, P. **Requirements Engineering: A Good Practice Guide**. 1st. ed. New York, NY, USA: John Wiley & Sons, Inc., 1997.

SOUZA LIMA, F. M. H. et al. A mixed load capacitated rural school bus routing problem with heterogeneous fleet: Algorithms for the Brazilian context. **Expert Systems with Applications**, v. 56, p. 320–334, 2016.

STÖCKER, D. **Java OpenStreetMap Editor**. Disponível em: <<https://josm.openstreetmap.de/>>. Acesso em: 5 set. 2019.

TIWARI, S. **Professional NoSQL**. Birmingham, UK, UK: Wrox Press Ltd., 2011.