



**UFOP**

Universidade Federal  
de Ouro Preto

**Universidade Federal de Ouro Preto  
Instituto de Ciências Exatas e Aplicadas  
Departamento de Computação e Sistemas**

**SBRP Management: Um Sistema de  
Informação para Gestão de Clientes  
no Contexto do Transporte Escolar  
Público**

**Raquel Martins dos Santos**

**TRABALHO DE  
CONCLUSÃO DE CURSO**

**ORIENTAÇÃO:**  
Rafael Frederico Alexandre

**Janeiro, 2022  
João Monlevade–MG**

**Raquel Martins dos Santos**

**SBRP Management: Um Sistema de  
Informação para Gestão de Clientes no  
Contexto do Transporte Escolar Público**

Orientador: Rafael Frederico Alexandre

Monografia apresentada ao curso de Sistemas de Informação do Instituto de Ciências Exatas e Aplicadas, da Universidade Federal de Ouro Preto, como requisito parcial para aprovação na Disciplina “Trabalho de Conclusão de Curso II”.

**Universidade Federal de Ouro Preto**

**João Monlevade**

**Janeiro de 2022**

## SISBIN - SISTEMA DE BIBLIOTECAS E INFORMAÇÃO

S237s Santos, Raquel Martins dos.  
SBRP Management [manuscrito]: um sistema de informação para  
gestão de clientes no contexto do transporte escolar público. / Raquel  
Martins dos Santos. - 2022.  
53 f.: il.: color., tab..

Orientador: Prof. Dr. Rafael Frederico Alexandre.  
Monografia (Bacharelado). Universidade Federal de Ouro Preto.  
Instituto de Ciências Exatas e Aplicadas. Graduação em Sistemas de  
Informação .

1. Aplicações web. 2. Engenharia de software. 3. Java (Linguagem de  
programação de computador). 4. Sistemas de recuperação da informação  
- Transportes. 5. Transporte escolar - ônibus. I. Alexandre, Rafael  
Frederico. II. Universidade Federal de Ouro Preto. III. Título.

CDU 004.775

Bibliotecário(a) Responsável: Flavia Reis - CRB6-2431



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE OURO PRETO  
REITORIA  
INSTITUTO DE CIÊNCIAS EXATAS E APLICADAS  
DEPARTAMENTO DE COMPUTAÇÃO E SISTEMAS



## FOLHA DE APROVAÇÃO

**Raquel Martins dos Santos**

**SBRP Management: Um Sistema de Informação para Gestão de Clientes no Contexto do Transporte Escolar Público**

Monografia apresentada ao Curso de Sistemas de Informação da Universidade Federal de Ouro Preto como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação

Aprovada em 10 de janeiro de 2022

Membros da banca

Doutor - Rafael Frederico Alexandre - Orientador (Universidade Federal de Ouro Preto)  
Doutor - Fernando Bernardes Oliveira - (Universidade Federal de Ouro Preto)  
Doutor - Bruno Pereira dos Santos - (Universidade Federal de Ouro Preto)

Rafael Frederico Alexandre, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 01/04/2022



Documento assinado eletronicamente por **Rafael Frederico Alexandre, PROFESSOR DE MAGISTERIO SUPERIOR**, em 01/04/2022, às 17:43, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site [http://sei.ufop.br/sei/controlador\\_externo.php?acao=documento\\_conferir&id\\_orgao\\_acesso\\_externo=0](http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0), informando o código verificador **0304444** e o código CRC **FBA63138**.

Referência: Caso responda este documento, indicar expressamente o Processo nº 23109.004106/2022-93

SEI nº 0304444

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35400-000  
Telefone: (31)3808-0819 - [www.ufop.br](http://www.ufop.br)

*Aos meus pais e a todos meus amigos que com muito apoio e carinho me incentivaram a enfrentar todos os obstáculos e adversidades da minha vida.*

# Agradecimentos

A todos os professores do curso, à Visão e à atlética. Obrigada por participarem dessa jornada pessoal e acadêmica. À república Ti-Ti-Ti que me apoiou em todas as etapas e foram essenciais para meu crescimento pessoal. À empresa ClubPetro pela oportunidade de primeira experiência profissional como programadora. Agradeço ao meu orientador Rafael, por ter me acompanhado nessa etapa, acreditar e incentivar na minha capacidade para desenvolver esse projeto e ter me guiado durante o decorrer deste trabalho. E agradeço também ao professor Fernando, pela ótima didática no meu primeiro semestre, que me incentivou a ter afinidade com a área de programação.

Por fim, a todos que contribuíram para meu crescimento pessoal e profissional, muito obrigada!

*“Se você ultrapassar aquela sensação de medo, aquele sentimento de estar correndo um risco, coisas verdadeiramente maravilhosas podem acontecer.”*

— Marissa Mayer

# Resumo

O transporte escolar, especialmente em pequenas cidades do Brasil, é um serviço que deve ser fornecido pelos órgãos públicos de forma eficiente e que procure atender a toda comunidade escolar dos municípios. Esse serviço envolve todo o planejamento de rotas dos veículos responsáveis pelo transporte dos alunos atendendo às restrições operacionais definidas pelos gestores municipais. Além disso, o acompanhamento da execução do que foi planejado para o município é fundamental para que os gestores públicos possam avaliar a qualidade do serviço. Para tanto, softwares que realizam o gerenciamento do transporte escolar são fundamentais para gestores municipais. Dado o elevado número de municípios brasileiros, segundo o Instituto Brasileiro de Geografia e Estatística [Instituto Brasileiro de Geografia e Estatística \(2022\)](#) são 5.570, gerir os contratos de licença para o uso de um software no contexto do transporte público escolar para todas essas unidades político-administrativas é uma tarefa complexa. Portanto, o objetivo pretendido neste trabalho é o desenvolvimento de um sistema web para gerenciar os contratos de serviços e reuniões de negócios para um sistema de roteamento de ônibus escolares. Durante o desenvolvimento foram utilizadas técnicas da engenharia de software que serviram de auxílio durante todo o processo. O sistema utiliza a linguagem *Typescript* e a biblioteca *React* juntamente com o banco de dados PostgreSQL e a linguagem Java para a construção de uma *Application Programming Interface (API)*. Com base nos resultados obtidos nos testes realizados após a conclusão do projeto, o sistema desenvolvido atendeu a demanda de gerenciar os contratos de uso de um software.

**Palavras-chaves:** Roteamento de Veículos Escolares. Transporte Público. Transporte Escolar. Web. Typescript. Java.

# Abstract

School transport, especially in small cities in Brazil, is a service that must be efficiently provided by public agencies and that serves the entire school community in the municipalities. This service involves the entire route planning of vehicles responsible for transporting students, taking into account the restrictions defined by municipal managers. In addition, monitoring the execution of what was planned for the municipality is essential for City Hall to be able to assess the quality of the service. Therefore, software that performs the management school transport is essential for municipal managers. Due to the high number of Brazilian municipalities, according to the Brazilian Institute of Geography and Statistics [Instituto Brasileiro de Geografia e Estatística \(2022\)](#) are 5,570, managing license contracts for the use of software in the context of public transport in schools for all political-administrative units is a complex task. Therefore, the intended objective of this work is the development of a web system to manage service contracts and business meetings for a school bus routing system. During development, software engineering techniques were used to help throughout the entire process. The system uses Typescript language and a React library together with a PostgreSQL database and a Java language to build an *Application Programming Interface* (API). Based on the results obtained in the tests carried out after the conclusion of the project, the developed system met the demand to manage the contracts for the use of a software.

**Key-words:** School Bus Routing. Public Transportation. School Transportation. Web. Typescript. Java.

# Lista de ilustrações

Figura 1 – Diagrama <i>SBR-Project</i> . . . . .	17
Figura 2 – Simples Agenda . . . . .	21
Figura 3 – Google Agenda . . . . .	22
Figura 4 – Sabe Contrato . . . . .	23
Figura 5 – ClickUp . . . . .	25
Figura 6 – Diagrama Casos de Uso . . . . .	26
Figura 7 – Modelagem <i>SBRP Management</i> . . . . .	28
Figura 8 – Modelagem <i>SBR-Project</i> . . . . .	29
Figura 9 – Organização de pastas do código <i>back-end</i> . . . . .	30
Figura 10 – API Restful . . . . .	31
Figura 11 – Virtualização da DOM . . . . .	32
Figura 12 – Login . . . . .	35
Figura 13 – Listagem de cidades . . . . .	36
Figura 14 – Remover cidade . . . . .	37
Figura 15 – Adicionar cidade . . . . .	38
Figura 16 – Editar cidade . . . . .	39
Figura 17 – Listagem de vendedores . . . . .	40
Figura 18 – Remover vendedor . . . . .	41
Figura 19 – Adicionar vendedor . . . . .	42
Figura 20 – Editar vendedor . . . . .	42
Figura 21 – Listagem de reuniões . . . . .	43
Figura 22 – Remover reunião . . . . .	44
Figura 23 – Adicionar reunião . . . . .	45
Figura 24 – Editar reunião . . . . .	46
Figura 25 – SBR-Project: Mapa com informações do banco de dados . . . . .	52
Figura 26 – SBR-Project: Mapa com os marcadores dos pontos de ônibus . . . . .	52
Figura 27 – SBR-Project: Mapa com os marcadores das residências dos alunos . . . . .	53
Figura 28 – SBR-Project: Informações sobre as rotas . . . . .	53

# Lista de tabelas

Tabela 1 – Sistemas correlatos . . . . .	24
Tabela 2 – Funcionalidades do Diagrama Casos de Uso . . . . .	27

# Lista de abreviaturas e siglas

**API** *Application Programming Interface*

**DOM** *Document Object Model*

**HTTP** *Hypertext Transfer Protocol*

**JSON** *JavaScript Object Notation*

**JWT** *JSON Web Token*

**MVC** *Model Controller View*

**SBRP** *School Bus Routing Project*

**SIG** *Sistema Integrado de Gestão*

**SPA** *Single Page Application*

**UFOP** *Universidade Federal de Ouro Preto*

**UML** *Unified Modeling Language*

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>16</b>
1.1	Justificativa	17
1.2	Objetivos	17
1.3	Estrutura do trabalho	18
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>19</b>
2.1	<i>School Bus Routing Problem</i>	19
2.2	Sistemas correlatos	20
2.2.1	Simple Agenda	20
2.2.2	Google Agenda	21
2.2.3	Sabe Contrato	22
2.3	Considerações finais	23
<b>3</b>	<b>DESENVOLVIMENTO</b>	<b>25</b>
3.1	Scrum	25
3.2	Casos de uso	26
3.3	Modelagem de dados	27
3.4	<i>Back-end</i>	30
3.5	<i>Front-end</i>	32
3.6	Considerações finais	33
<b>4</b>	<b>RESULTADOS</b>	<b>34</b>
4.1	Login	34
4.2	Cidades	35
4.2.1	Listagem de cidades	36
4.2.2	Remover cidade	37
4.2.3	Adicionar cidade	38
4.2.4	Editar cidade	39
4.3	Vendedores	39
4.3.1	Listagem de vendedores	40
4.3.2	Remover vendedor	41
4.3.3	Adicionar vendedor	42
4.3.4	Editar vendedor	42
4.4	Reuniões	43
4.4.1	Listagem de reuniões	43
4.4.2	Remover reunião	44

4.4.3	Adicionar reunião . . . . .	45
4.4.4	Editar reunião . . . . .	46
<b>4.5</b>	<b>Considerações finais . . . . .</b>	<b>46</b>
<b>5</b>	<b>CONCLUSÃO . . . . .</b>	<b>47</b>
5.0.1	Considerações finais . . . . .	47
5.0.2	Trabalhos futuros . . . . .	47
	<b>REFERÊNCIAS . . . . .</b>	<b>49</b>
	<b>APÊNDICES</b>	<b>51</b>
	<b>APÊNDICE A – TELAS DO SISTEMA SBR-PROJECT . . . . .</b>	<b>52</b>

# 1 Introdução

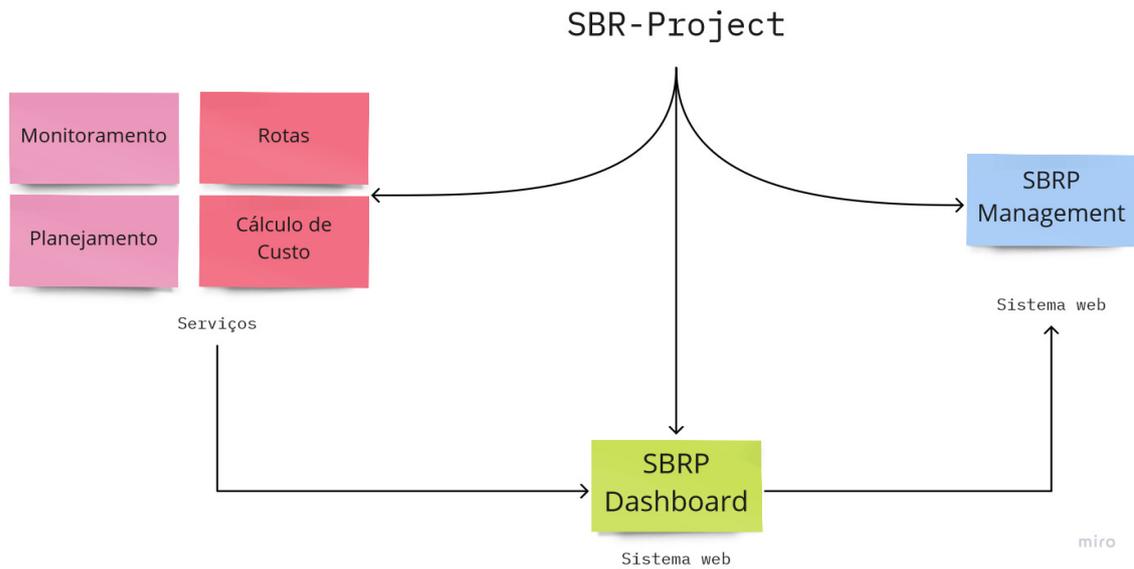
O acesso ao transporte público é fundamental para facilitar e garantir a permanência de alunos na escola, tanto os que residem em áreas rurais quanto os que moram em áreas urbanas. Fornecer o transporte escolar é uma forma de contribuir para a redução da evasão escolar, uma vez que muitos alunos acabam abandonando o estudo porque não conseguem chegar à escola (MENEZES; OLIVEIRA, 2019). De acordo com o [Fundo Nacional de Desenvolvimento da Educação \(2019\)](#) a Lei nº 9.394/96 da Constituição Federal de 1988 prevê o direito do aluno de escola pública o uso do transporte escolar, mediante a obrigação de estado e municípios.

O Problema de Roteamento de Ônibus Escolares, no termo inglês *School Bus Routing Problem*, consiste em minimizar o custo total de roteamento para pegar estudantes e entregá-los a uma escola, satisfazendo a capacidade do veículo e a distâncias máxima que estudantes podem caminhar de sua casa até as paradas de ônibus (NETO et al., 2020). Sendo assim, para solucionar este problema, está sendo desenvolvido pelos alunos da Universidade Federal de Ouro Preto (UFOP) um sistema *web* para roteamento de veículos escolares, intitulado *SBR-Project*. As telas desse sistema são apresentadas no [Apêndice A](#).

O *SBR-Project* é um projeto extenso com diversos módulos desenvolvidos, a modelagem de dados do projeto inteiro será apresentado na [Figura 1](#). Entre os módulos, [Lima \(2021\)](#) desenvolveu um módulo *web* para rastreamento e monitoramento de veículos de transporte escolar em ambientes reais. [Araújo \(2021\)](#) elaborou uma metodologia para realizar o cálculo do custo monetário de cada rota propostas por algoritmos de otimização. E [Messias \(2018\)](#) desenvolveu um sistema de informação *web* para visualização de rotas em ambientes reais. Para utilizar o *SBR-Project* será necessário obter uma licença de uso, formalizada através de um contrato envolvendo o gerente do sistema e o membro da prefeitura de uma cidade. Logo, o desenvolvimento de um módulo de gerenciamento se faz necessário para administrar os contratos de serviços e reuniões de apresentação do *SBR-Project* para o cliente. O *SBRP*<sup>1</sup> *Management* é um sistema *web* desenvolvido pela autora deste trabalho para atender essa demanda, sendo de grande valia para os gerentes do projeto devido a facilidade de acesso às informações sobre o andamento do processo de contratação da plataforma pelos clientes.

---

<sup>1</sup> *School Bus Routing Project (SBRP)*

Figura 1 – Diagrama *SBR-Project*

Fonte: produzido pela autora

Como solução para atender a demanda do *SBR-Project* surgiu a necessidade de um Sistema Integrado de Gestão (SIG), que fornecerá uma estrutura para registrar as cidades, contratos e vendedores, podendo assim editar informações, cadastrar contratos e marcar reuniões de negócios. Portanto, com a criação da aplicação proposta neste trabalho, os gerentes dos sistemas poderão controlar de maneira apropriada os contratos de serviços.

## 1.1 Justificativa

Com o objetivo de desenvolver uma plataforma de gerenciamento de clientes, o *SBRP Management* é um sistema *web* separado da plataforma principal, compartilhando apenas a mesma base de dados. O sistema proposto neste projeto se faz importante para o *SBR-Project*, principalmente por agilizar o processo de contratação para os gestores dado que através dele é possível acompanhar os contratos em andamento. Levando em consideração que a plataforma utilizará um SIG é esperado que a implantação da aplicação contribua positivamente para a melhoria do processo atual.

## 1.2 Objetivos

O objetivo principal deste trabalho é criar um ambiente que permita o acompanhamento de contratos e que realize o gerenciamento envolvendo o processo de contratação de serviço, atendendo a demanda problematizada na Seção anterior.

Os objetivos específicos têm como finalidade atender as demandas do gestor para o acompanhamento do processo de contratação através do sistema *web*. Sendo eles:

- **Visualização dos registros das reuniões de negócios:** a leitura de dados em tabelas é fundamental para o gestor pois permite analisar quais contratos foram finalizados, estão em débito ou foram recusados pelo cliente na negociação que, nesse caso, seriam os municípios que utilizam os serviços do *SBR-Project*.
- **Auxiliar na tomada de decisão:** permite o gestor observar os dados e motivos de recusa do contrato e, assim, planejar soluções para melhorar as estratégias de venda. Segundo [Guimarães e Évora \(2004\)](#), a tomada de decisão é considerada a função que caracteriza o desempenho da gerência e, independentemente do aspecto da decisão, esta atitude deve ser fruto de um processo sistematizado, que envolve o estudo do problema a partir de um levantamento de dados, produção de informação, estabelecimento de propostas de soluções, escolha da decisão, viabilização e implementação da decisão e análise dos resultados obtidos.
- **Facilitar o acompanhamento do processo de negociação dos gestores com os clientes:** além do usuário visualizar os dados armazenados, terá a opção de editar informações e registrar reuniões de negócios.

### 1.3 Estrutura do trabalho

O restante deste trabalho é organizado como segue. No [Capítulo 2](#) é apresentada a fundamentação teórica do trabalho desenvolvido, as ferramentas, as tecnologias e os sistemas relacionados. O [Capítulo 3](#) descreve os requisitos e a modelagem do sistema. No [Capítulo 4](#) são apresentados os detalhes da implementação, funcionalidades e metodologia utilizada no desenvolvimento do projeto, além dos resultados obtidos. Por fim, são apresentadas as considerações finais e propostas de trabalhos futuros no [Capítulo 5](#).

## 2 Revisão bibliográfica

Este Capítulo apresenta, a revisão literária que foi utilizada neste trabalho e os sistemas de informação que apresentam similaridades com o projeto desenvolvido. A [Seção 2.1](#) apresenta a revisão literária sobre o *School Bus Routing Problem* e na [Seção 2.2](#) são apresentados os sistemas correlatos.

### 2.1 *School Bus Routing Problem*

O *School Bus Routing Problem* tem sido constantemente estudado desde o surgimento da primeira publicação sobre ele por [Newton e Thomas \(1969\)](#). De acordo com [Desrosiers J. \(1981\)](#), o *School Bus Routing Problem* pode ser resolvido por cinco etapas: preparação de dados, seleção do ponto de ônibus (atribuição do aluno a paradas), geração de rota de ônibus, ajuste do horário do sino da escola e programação da rota.

1. **Preparação de dados:** os dados escolares contêm informações sobre a localização do escolas, o horário de início e término das escolas para chegadas de ônibus, e o tempo máximo de viagem de um aluno em um ônibus.
2. **Seleção de parada de ônibus:** a seleção de ponto de ônibus busca selecionar um conjunto de pontos de ônibus e atribuir alunos a essas paradas. Para escolas em áreas rurais, os alunos são geralmente recolhidos em suas casas. No entanto, em áreas urbanas áreas, presume-se que os alunos caminhem até um ponto de ônibus de suas casas e pegar um ônibus na parada.
3. **Geração de rota de ônibus:** na geração de rotas de ônibus, são construídas as rotas escolares. Depois que as rotas da escola são geradas, heurísticas de melhoria podem ser aplicado nas rotas.
4. **Programação de rota:** a programação da rota especifica a hora exata de início e término de cada rota e forma uma cadeia de rotas que podem ser executadas sucessivamente pelo mesmo ônibus.

[Park e Kim \(2010\)](#) abordaram outras questões importantes a se considerar ao implementar uma solução para o *School Bus Routing Problem* que estão listadas a seguir:

- **Problema da manhã e da tarde:** na manhã, um ônibus pega os alunos nos pontos de ônibus para deixá-los. E no trajeto da volta, os alunos são buscados nas escolas e entregues em seus pontos de ônibus no período da tarde. [Braca J. \(1997\)](#) afirmou

que o problema da manhã é mais difícil do que o problema da tarde por dois motivos: o horário de início das aulas matinais entre as escolas são mais dispersos e tráfego urbano é mais intenso.

- **Transporte de alunos com necessidades especiais:** os alunos com necessidades especiais são buscados diretamente em suas casas e não em seus pontos de ônibus.

## 2.2 Sistemas correlatos

Nesta Seção serão apresentados *softwares* semelhantes com características fundamentais para a gestão de contratos de serviço do *SBR-Project* e que serão desenvolvidas pelo sistema proposto nesse trabalho. As funcionalidades mais importante para o gestor é o acompanhamento das datas de reuniões e a possibilidade de anexar o contrato de cada cliente, por isso os sistemas correlatos têm foco em *softwares* de calendário que permitem anexar arquivos.

### 2.2.1 Simples Agenda

O Simples Agenda é um sistema *White Label* com um nicho de mercado bem diversificado como salões de beleza, consultórios médicos e estúdios de fotografia. Oliveira (2021) definiu como *White Label* a terceirização de desenvolvimento de produtos e serviços em que se produz um molde padrão que resolva problemas comuns de diversas empresas, bastando que a parte interessada personalize o modelo com sua identidade visual.

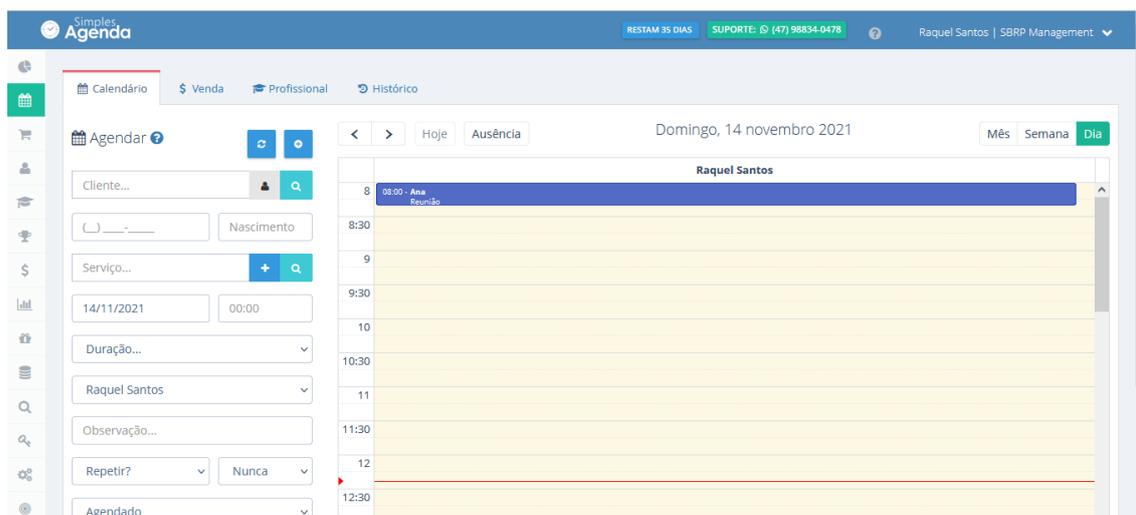
O Simples Agenda é uma ferramenta brasileira desenvolvida em 2016 pela Puglisi Desenvolvimento de Sistemas que oferece total controle sobre os agendamentos dos seus clientes, melhor gestão de tempo e aumento da produtividade graças à automação. A documentação está disponível no site oficial<sup>1</sup>.

Alguns recursos disponíveis são visualização dos eventos no calendário, emissão de nota fiscal e aplicativo para dispositivos móveis. A Figura 2 mostra a tela principal do sistema.

---

<sup>1</sup> <<https://www.simplesagenda.com.br/site/Funcionalidades/agenda-online.php>>

Figura 2 – Simple Agenda



Fonte: produzido pela autora

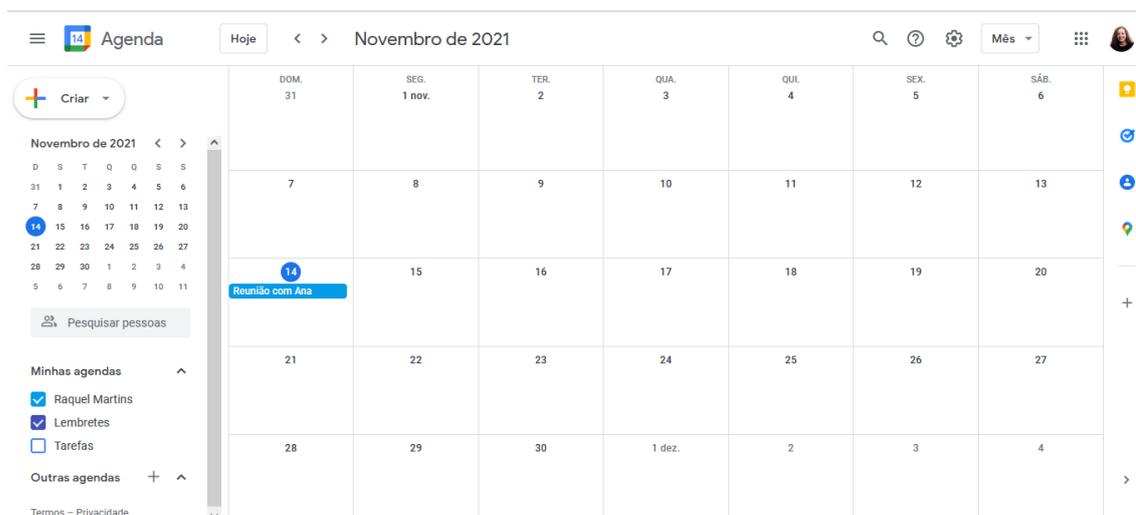
Este *software* foi cogitado para gerenciar o agendamento das reuniões do *SBR-Project*, mas, além de existir um custo mensal para o uso, não contempla os requisitos de registro de *feedback* das reuniões, anexo de arquivo de contrato, visualização de dados em tabela e ordenação por status de venda. Por esses recursos serem essenciais para auxiliar na tomada de decisão do gerente do sistema, o Simple Agenda foi desconsiderado para cumprir a demanda proposta.

### 2.2.2 Google Agenda

De acordo com Feddern-Bekcan (2008), o Google Agenda <sup>2</sup> é um calendário online gratuito e fácil de usar para uso pessoal ou no local de trabalho, como listar eventos e coordenar reuniões, conforme mostra a Figura 3.

<sup>2</sup> Disponível em: <<https://workspace.google.com/intl/pt-BR/products/calendar>>

Figura 3 – Google Agenda



Fonte: produzido pela autora

As principais vantagens dessa plataforma é a interface ser intuitiva, gratuita, possuir um aplicativo para dispositivos móveis e estar integrada com o Google Meet.

O Meet é uma plataforma gratuita que oferece segurança e proteção em videochamadas para empresas, sua documentação está disponível do site oficial<sup>3</sup>. O uso dessa ferramenta é especialmente importante devido ao notável aumento do trabalho remoto no Brasil. Dados coletados pelo Instituto de Pesquisa Econômica Aplicada (IPEA, 2021) relata que 11% dos brasileiros se encontram nesse modelo de trabalho em 2021.

Caso não houvesse a necessidade de detalhar as informações sobre as reuniões e cidades envolvidas, o Google Agenda poderia ser uma solução para o problema proposto, devido ao fato do sistema ser gratuito.

Apesar do *SBRP Management* não oferecer integração com os serviços da Google, existe a possibilidade de registrar em cada reunião um link para a videochamada do Google Meet caso a reunião ocorra de forma remota, já que os vendedores e clientes não necessariamente moram em regiões próximas.

### 2.2.3 Sabe Contrato

A Sabe Sistemas<sup>4</sup> é uma empresa brasileira que está a mais de 10 anos no mercado de softwares para gestão empresarial. Os sistemas disponíveis que se encaixam no modelo de negócio proposto são:

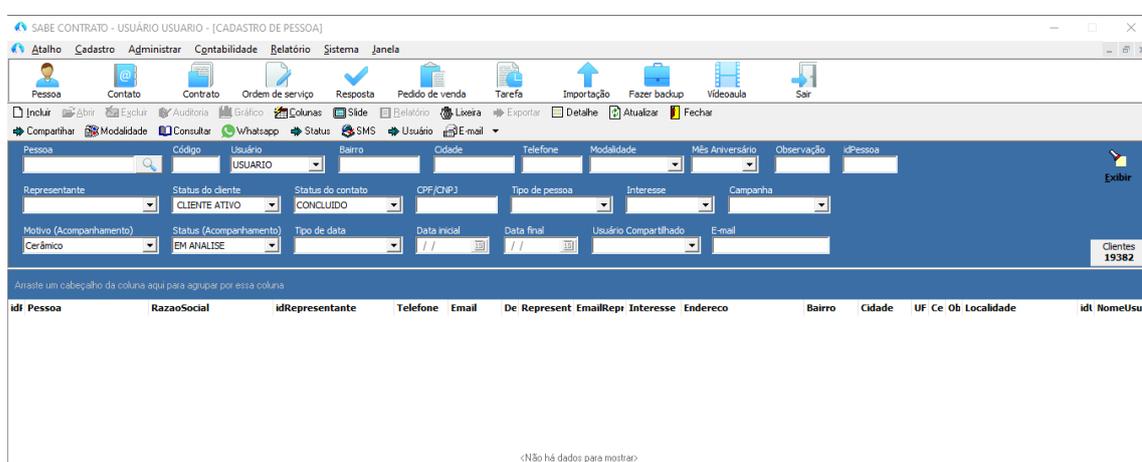
<sup>3</sup> Disponível em: <<https://workspace.google.com/intl/pt-BR/products/meet>>

<sup>4</sup> Disponível em: <<https://sabisistemas.com.br/empresa-desenvolvimento-sistema-programa>>

- Sabe Contrato
- Sabe CRM
- Sabe Nota Fiscal
- Sabe Gestão Integrada

Entre esses sistemas, o Sabe Contrato<sup>5</sup> é um sistema *desktop* bem completo que oferece alguns recursos como: visualização de dados em tabela, ordenação por status, armazenamento dos arquivos de contrato, envio de *e-mails* e integração com o Skype e WhatsApp. É possível ver a tela principal do sistema na Figura 4.

Figura 4 – Sabe Contrato



Fonte: produzido pela autora

Para a realização de testes, foi instalado a versão gratuita e notou-se que o sistema não oferece uma boa experiência ao usuário devido a travamentos e interface confusa. No site não há tutoriais de como usar a plataforma e nem informações sobre o custo da licença de uso. Sendo assim, não foi considerado como opção para suprir a demanda proposta neste trabalho.

## 2.3 Considerações finais

Neste Capítulo foram apresentadas as tecnologias e ferramentas utilizadas para o desenvolvimento do *SBRP Management* e os sistemas correlatos. O estudo sobre sistemas correlatos foram importantes para auxiliar no planejamento de requisitos. Na Tabela 1 é possível ver um comparativo entre os sistemas apresentados.

<sup>5</sup> Disponível em: <<https://sabisistemas.com.br/sabe-contrato-software>>

Tabela 1 – Sistemas correlatos.

<b>Funcionalidades</b>	<b>Simple Agenda</b>	<b>Google Agenda</b>	<b>Sabe Contrato</b>	<b>SBRP Management</b>
Visualização de calendário	x	x	-	-
Criação de eventos	x	x	-	x
Comentários dos participantes de um evento	x	-	-	x
Download e upload de arquivos	-	-	x	x
Visualização de dados em tabela	-	-	x	x
Aplicativo para dispositivos móveis	x	x	-	-
Baixo custo	-	x	-	x
Organização de arquivos	-	-	-	x

Como observado na tabela acima, o *SBRP Management* apenas não atendeu os requisitos de visualização de calendário e possuir um aplicativo para dispositivos móveis. Mas em contraponto, as tabelas de reuniões do sistema apresentam a opção de ordenação por data em ordem crescente ou decrescente, sendo possível identificar rapidamente os eventos que estão próximos de acontecer. E também, apesar de não ter um aplicativo, o site está responsivo, ou seja, a tela se adapta ao tamanho do dispositivo no qual está sendo acessado.

No [Capítulo 3](#) serão descritos os requisitos da plataforma e a metodologia utilizada para implementar as funcionalidades propostas.

## 3 Desenvolvimento

Neste Capítulo é detalhado o desenvolvimento do projeto. Na [Seção 3.1](#) é descrita a metodologia ágil utilizada para o desenvolvimento do sistema. A [Seção 3.2](#) são os casos de uso. A modelagem do banco de dados é apresentado na [Seção 3.3](#). Na [Seção 3.4](#) é apresentado o *back-end*. Por fim, na [Seção 3.5](#) são as ferramentas e métodos para o *front-end* e na [Seção 3.6](#) as considerações finais.

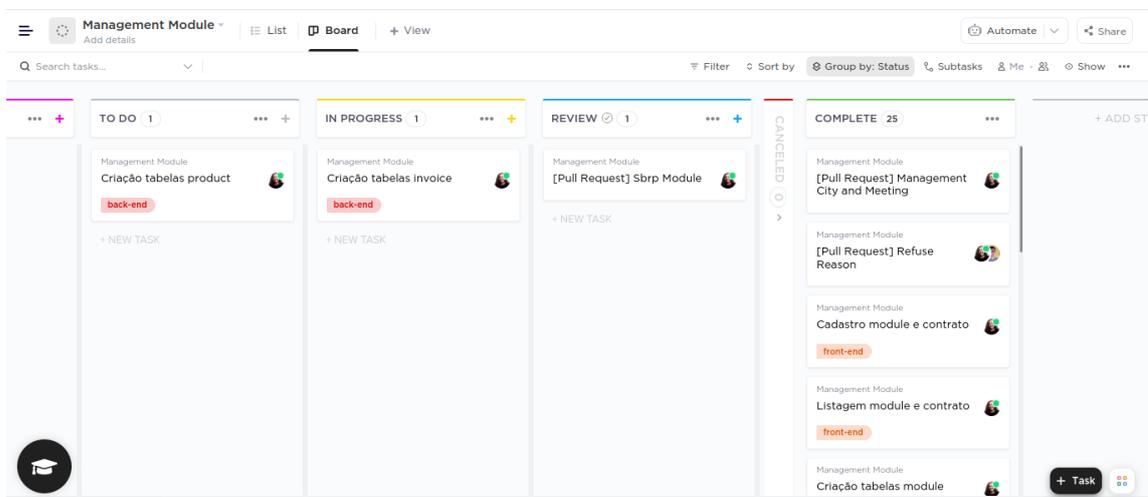
### 3.1 Scrum

De acordo com o [Soares \(2004\)](#) a idéia principal do *Scrum* é que o desenvolvimento de softwares envolve muitas variáveis técnicas e do ambiente, como requisitos, recursos e tecnologia que podem mudar durante o processo, tornando o processo de desenvolvimento imprevisível e complexo, requerendo flexibilidade para acompanhar as mudanças.

Neste projeto, a cada quatro dias foram feitas reuniões de *Daily Scrum* devido ao fato de a equipe ser formada apenas pela autora deste trabalho e o orientador que assumiu o papel de *Product Owner*. As reuniões de *sprint* foram feitas a cada quinzena para demonstração do desenvolvimento, possíveis impedimentos que ocorreram neste período e planejamento de novas atividades do ciclo seguinte.

Para organização das tarefas a serem desenvolvidas foi utilizado a plataforma ClickUp, conforme mostra a [Figura 5](#).

Figura 5 – ClickUp



Fonte: produzido pela autora

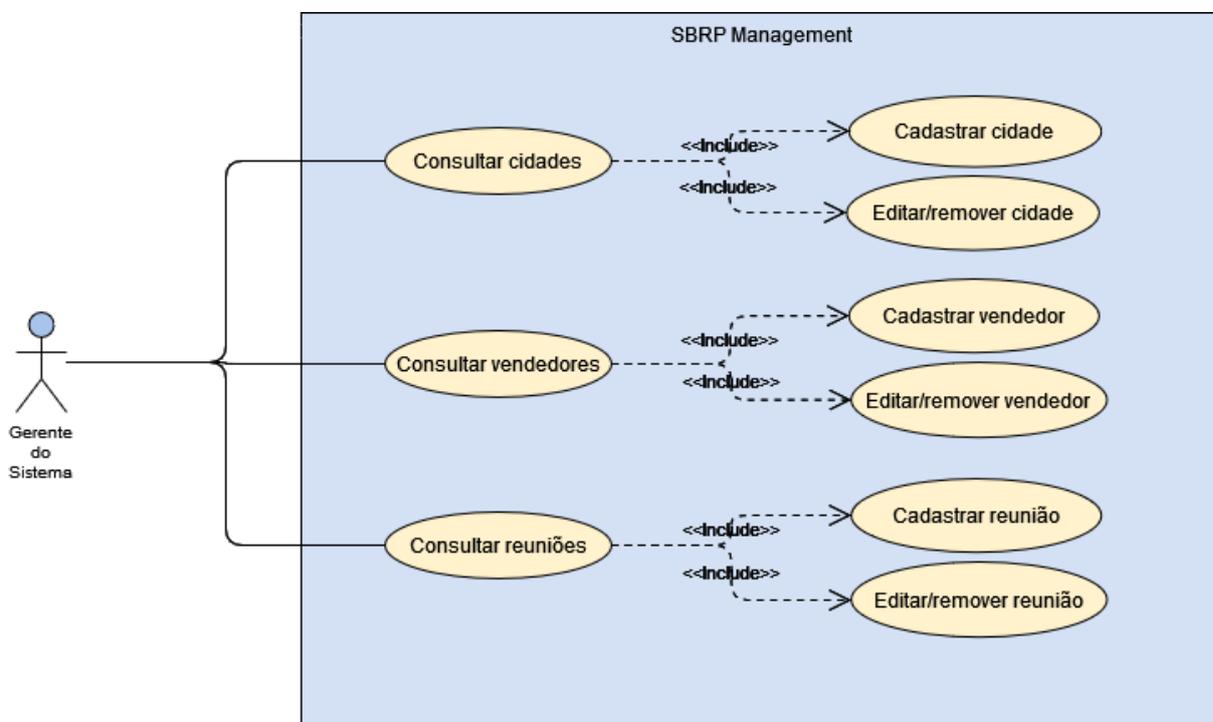
As tarefas cadastradas ClickUp foram organizadas da seguinte forma:

1. **TO DO:** tarefas planejadas na reunião de *sprint* que não foram inicializadas.
2. **IN PROGRESS:** tarefas em andamento.
3. **REVIEW:** tarefas realizadas que necessitam de validação pelo *Product Owner*.
4. **COMPLETE:** tarefas que já foram testadas e validadas.

## 3.2 Casos de uso

Para o levantamento de requisitos foi utilizado o diagrama de casos de uso da *Unified Modeling Language (UML)*. De acordo com Pressman e Maxim (2016), um caso de uso identifica os atores envolvidos em uma interação e dá nome ao tipo de interação. Ainda segundo esses autores, o caso de uso é uma técnica eficaz para elicitare requisitos dos *stakeholders* que vão interagir diretamente com o sistema. A Figura 6 mostra os casos de uso do *SBRP Management*.

Figura 6 – Diagrama Casos de Uso



Fonte: produzido pela autora

A elaboração do diagrama de casos de uso foi essencial para o levantamento de requisitos e também para os primeiros esboços do layout das telas da aplicação. Ao

final do desenvolvimento, esse diagrama foi consultado novamente para garantir que todas demandas planejadas foram devidamente cumpridas. Para maior detalhamento das operações que o gerente pode executar, a [Tabela 2](#) mostra as funcionalidades do sistema.

Tabela 2 – Funcionalidades do Diagrama Casos de Uso.

<b>Código</b>	<b>Funcionalidade</b>	<b>Descrição</b>
<b>UC001</b>	Consultar cidades	Listagem de cidades e exibição dos status de contrato
<b>UC002</b>	Remover cidade	Remover cidade
<b>UC003</b>	Cadastrar cidade	Adicionar status, gerentes e contratos
<b>UC004</b>	Editar cidade	Editar status, gerentes e contratos
<b>UC005</b>	Consultar vendedores	Listagem de vendedores e as cidades onde atuam
<b>UC006</b>	Remover vendedor	Remover vendedor
<b>UC007</b>	Cadastrar vendedor	Cadastrar dados pessoais e cidades
<b>UC008</b>	Editar vendedor	Editar dados pessoais e cidades
<b>UC009</b>	Consultar reuniões	Listagem de reuniões e status
<b>UC010</b>	Remover reunião	Remover reunião
<b>UC011</b>	Cadastrar reunião	Adicionar vendedor, cidade, participantes e data
<b>UC012</b>	Editar reunião	Editar vendedor, cidade, participantes e data

- **UC001, UC002, UC003 e UC004:** as operações envolvendo cidades têm como função informar a quantidade de contratos, a data inicial e final de cada contrato, a situação financeira, a situação de venda e os gerentes responsáveis de uma cidade.
- **UC005, UC006, UC007 e UC008:** as operações envolvendo vendedores têm como função informar o nome, telefone, *e-mail* e as cidades de atuação de um vendedor.
- **UC009, UC010, UC011 e UC012:** as operações envolvendo reuniões têm como função informar a data, os participantes, o vendedor responsável, a decisão individual de cada participante e a decisão final de cada reunião.

Sendo assim, os requisitos do sistema apresentados têm como objetivo definir o que o sistema deve ser capaz de fazer e exibir as informações sobre contratos e cidades de forma clara ao usuário.

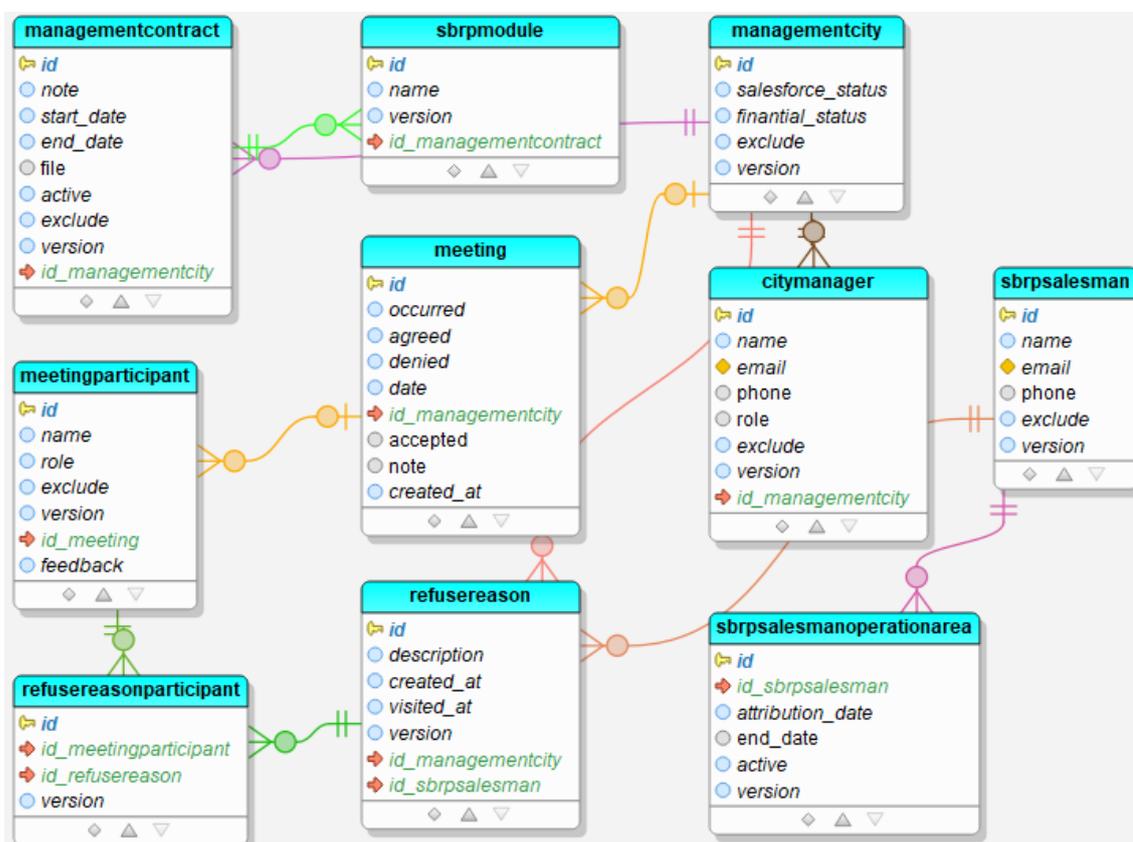
### 3.3 Modelagem de dados

Segundo [Pressman e Maxim \(2016\)](#), a modelagem de sistema é o processo de desenvolvimento de modelos abstratos de um sistema, em que cada modelo apresenta uma visão ou perspectiva, diferente do sistema. A modelagem de sistema geralmente representa o sistema com algum tipo de notação gráfica, que, atualmente, quase sempre é baseada em notações de [UML](#).

Para a elaboração da modelagem do banco de dados em diagramas foi usado o *software* pgModeler<sup>1</sup>. A modelagem do *SBRP Management* representada na Figura 7 inclui quatro principais entidades, são elas:

1. **Cidades:** cidades-alvo onde há o interesse de implantação do sistema.
2. **Vendedores:** responsáveis por oferecer a venda do sistema a uma cidade.
3. **Participantes:** representantes das cidades que participarão da reunião.
4. **Reuniões:** reunião para a apresentação do serviço, contém a cidade, vendedores e participantes.

Figura 7 – Modelagem *SBRP Management*

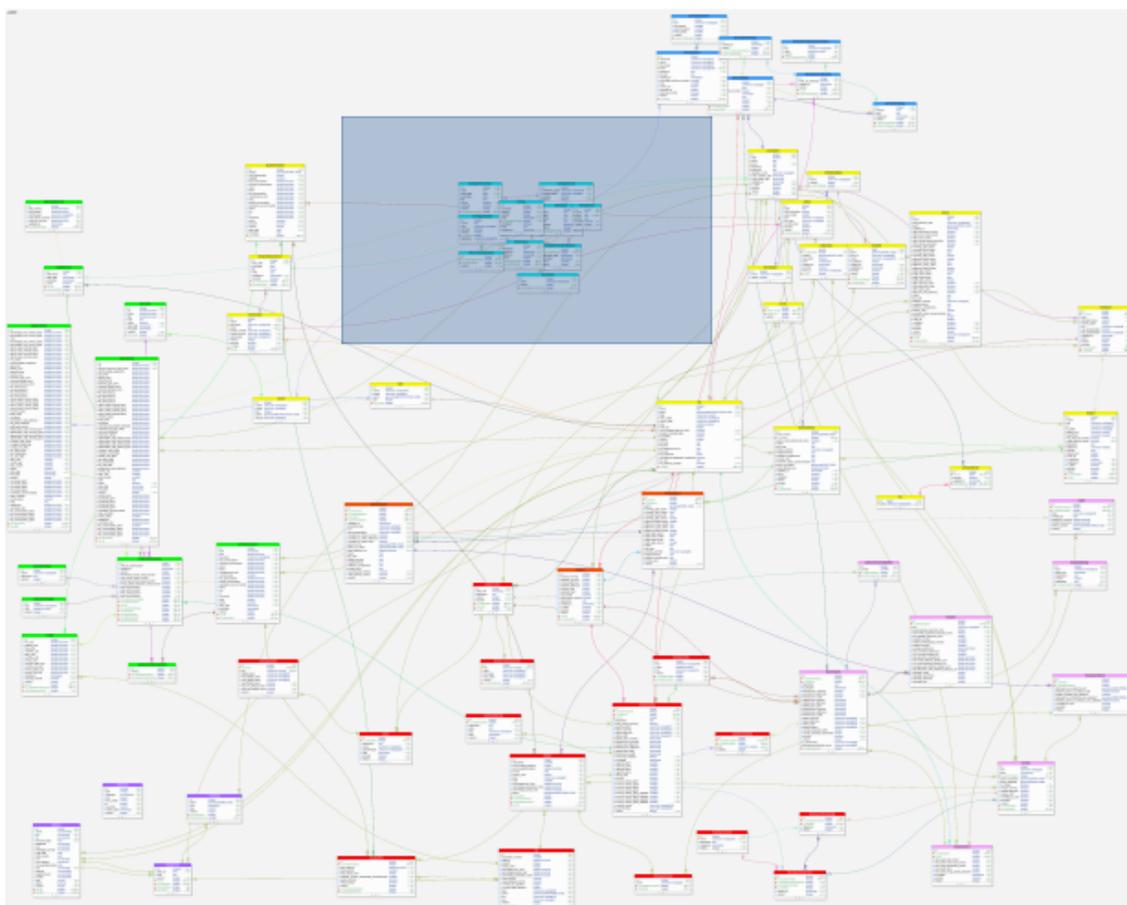


Fonte: produzido pela autora

<sup>1</sup> Disponível em: <<https://pgmodeler.io/>>

O *SBRP Management* faz parte do *SBR-Project*, no qual outros módulos foram desenvolvidos. A modelagem do projeto inteiro é apresentada na [Figura 8](#), sendo o retângulo em destaque as entidades do *SBRP Management*.

Figura 8 – Modelagem *SBR-Project*



Fonte: produzido pela autora

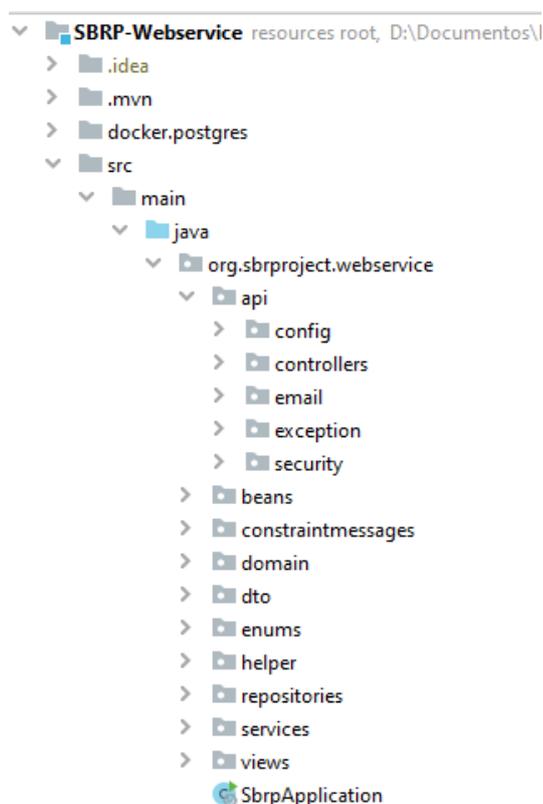
## 3.4 Back-end

O *back-end* dessa aplicação foi feito na linguagem Java com a combinação dos seguintes *frameworks*:

- **Spring<sup>2</sup>**: Injeção de dependências.
- **Hibernate<sup>3</sup>**: Mapeamento do objeto-relacional.

O padrão de projeto utilizado para o desenvolvimento foi o *Model Controller View (MVC)*. O objetivo deste padrão é separar o projeto em três camadas independentes, que são o modelo, a visão e o controlador. Essa separação de camadas ajuda na redução de acoplamento e promove o aumento de coesão nas classes do projeto. Assim, quando o modelo MVC é utilizado, pode facilitar a manutenção do código e sua reutilização em outros projetos (MEDEIROS, 2013). A Figura 9 representa a organização de pastas do *back-end*.

Figura 9 – Organização de pastas do código *back-end*



Fonte: produzido pela autora

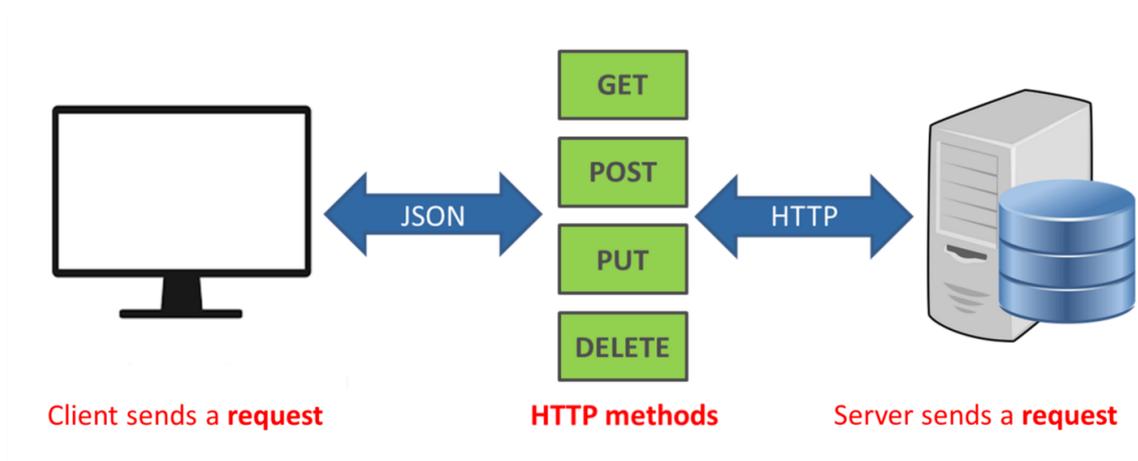
<sup>2</sup> Disponível em: <<https://spring.io>>

<sup>3</sup> Disponível em: <<https://hibernate.org>>

Para o processamento de dados e para facilitar o acesso para o *front-end* foi implementada uma *API Restful*. Segundo Antunes (2019) a *API Restful* é uma interface que fornece dados em um formato padronizado baseado em requisições *Hypertext Transfer Protocol (HTTP)*. Elas utilizam verbos para definir qual é a finalidade da requisição que está sendo enviada, a Figura 10 mostra a comunicação entre o cliente e o servidor através dos métodos *HTTP*, os verbos são apresentados na listagem a seguir:

- **GET:** a requisição é um pedido de dados para a API. A API vai buscar os dados solicitados em algum banco e, provavelmente, vai retornar em formato *JavaScript Object Notation (JSON)*
- **POST:** tipo de requisição utilizada para criar um recurso em uma determinada API. São chamados de recursos o objeto que está sendo tratado naquela API.
- **PUT:** requisição utilizada para atualizar o recurso indicado com alguma informação.
- **PATCH:** requisição feita para atualização de somente uma parte de um recurso.
- **DELETE:** requisição para excluir um dado.

Figura 10 – API Restful



Fonte: Benharosh (2018)

### 3.5 Front-end

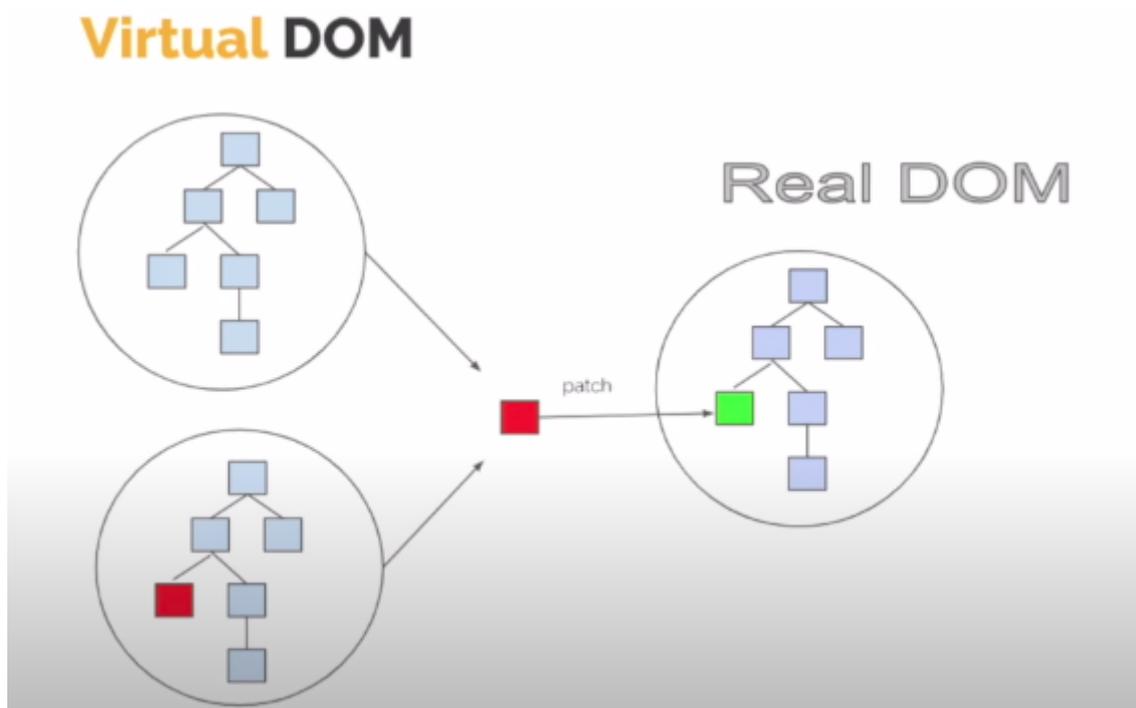
O *front-end* do sistema foi feita utilizando a linguagem *Typescript*<sup>4</sup> e a biblioteca *React*<sup>5</sup>. A escolha dessas tecnologias foram feitas devido a facilidade de separar os componentes visuais da parte lógica e, assim, tornando o código reaproveitável e organizado.

Outra vantagem relevante do *React* é a possibilidade de desenvolver aplicações *Single Page Application (SPA)* que são sites capazes de atualizar apenas um ou mais componentes de uma página, oferecendo ao usuário uma boa experiência semelhante a de aplicações *desktops*.

De acordo com [Aggarwal \(2018\)](#) uma das grandes vantagens da biblioteca *React* é o uso da *Document Object Model (DOM)* virtualizada, que é armazenada em memória. Sendo assim, quando há requisição para uma mudança é comparado a DOM do navegador e a virtualizada, e as mudanças são refletidas apenas onde está sendo usado, evitando uma renderização de toda a DOM e há ganhos altos em performance, especialmente quando existe um grande fluxo de dados.

A renderização apenas em um componente é ilustrado na [Figura 11](#).

Figura 11 – Virtualização da DOM



Fonte: [Wahyu \(2020\)](#)

<sup>4</sup> Disponível em: <<https://www.typescriptlang.org>>

<sup>5</sup> Disponível em: <<https://pt-br.reactjs.org>>

*Stakeholders* segundo Pressman e Maxim (2016) é qualquer pessoa que se beneficie de forma direta ou indireta do sistema que está sendo desenvolvido. Neste projeto, o gerente do sistema é o principal *stakeholder*, sendo ele responsável por administrar os contratos de serviços do *SBR-Project*. Dessa forma, as telas foram desenvolvidas com foco de apresentar de maneira sucinta ao gerente informações sobre as próximas cidades que podem aderir a plataforma.

## 3.6 Considerações finais

Nesse Capítulo foram discutidas a metodologia ágil utilizada para o desenvolvimento do sistema. E também as tecnologias usadas para o desenvolvimento do *back-end* e *front-end*. O Capítulo 4 apresentará o projeto e descrições das telas desenvolvidas.

## 4 Resultados

O presente Capítulo apresenta a utilização do sistema. Em cada Seção será apresentado o funcionamento de uma tela bem como a descrição de sua funcionalidade. Ao final, é apresentado as considerações finais.

Com o objetivo de internacionalizar o projeto, todas as telas foram desenvolvidas com o suporte de tradução para português ou inglês através da biblioteca *i18next*<sup>1</sup>. Por motivo de clareza, todas as imagens deste Capítulo estarão em português.

Neste Capítulo os componentes serão apresentados considerando o fluxo de navegação do usuário. A Seção 4.1, apresentará a tela de *login*. Já a Seção 4.2 apresentará as telas relacionadas as cidades cadastradas no sistema. A Seção 4.3 irá apresentar as telas relacionadas aos vendedores e na Seção 4.4 serão apresentadas as telas de reuniões. Por fim, a Seção 4.5 apresentam as considerações finais deste Capítulo.

### 4.1 Login

Apenas usuários com o acesso de gestor são autorizados a navegar pelo *SBRP Management*, a verificação é feita pelo *back-end*. Caso o usuário for válido, ele é redirecionado para a tela inicial do sistema.

Para a segurança dos dados, foi usado a autenticação do *JSON Web Token (JWT)* em todas as requisições para permitir apenas que usuários autorizados tenham acesso ao sistema e às requisições do *back-end*. A verificação do usuário é feita na tela de login e, caso o *login* do gestor e a senha estiverem corretos, é atribuído um *token* para autorizar o acesso às telas do sistema. De acordo com a documentação oficial do JWT<sup>2</sup>, ele é um *token* de acesso de forma compacta e independente para transmitir informações com segurança entre as partes com um objeto *JSON* e a autenticação de usuários é o cenário mais comum para o uso.

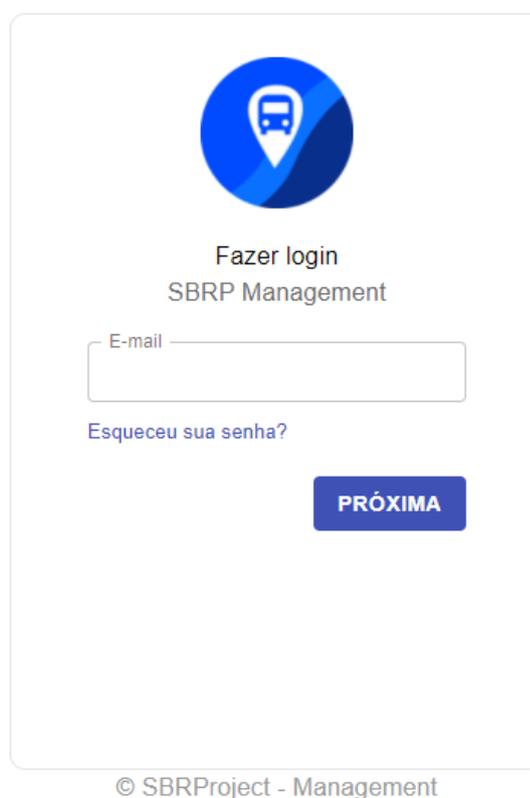
A Figura 12 mostra a tela de *login* do sistema onde solicita ao usuário que digite o e-mail e ao clicar em *PRÓXIMA* é solicitado a senha para autenticação de usuário.

---

<sup>1</sup> Disponível em: <<https://react.i18next.com/>>

<sup>2</sup> Disponível em: <<https://jwt.io/introduction>>

Figura 12 – Login



The image shows a login interface for 'SBRP Management'. At the top center is a blue circular logo containing a white bus icon. Below the logo, the text 'Fazer login' and 'SBRP Management' is displayed. A text input field is labeled 'E-mail'. Below the input field, there is a blue link that says 'Esqueceu sua senha?'. A blue button with the text 'PRÓXIMA' is positioned to the right of the input field. At the bottom center, the copyright notice '© SBRProject - Management' is visible.

Fonte: produzido pela autora

## 4.2 Cidades

Esta Seção apresenta as telas referente às cidades que podem aderir ao *SBR-Project*. É uma dos componentes mais importantes do sistema para o gestor, pois permite o acompanhamento da situação de venda do *SBR-Project* nas cidades, a inserção de novos contratos e alterar contratos já existentes.

### 4.2.1 Listagem de cidades

A listagem de cidades cadastradas no sistema é mostrada na [Figura 13](#).

Figura 13 – Listagem de cidades

Nome	Status de venda	Qtde contratos	Status financeiro	Opções
ABAETÉ - MG	Implantado	2	Débito	
TAPIRA - MG	Recusou	0	Débito	
AÇUCENA - MG	Recusou	0	Débito	
ARAXÁ - MG	Recusou	0	Débito	
ALMENARA - MG	Em análise	0	Sem débitos	
JOÃO MONLEVADE - MG	Em análise	0	Sem débitos	

Rows per page: 100 1-27 of 27 < >

Fonte: produzido pela autora

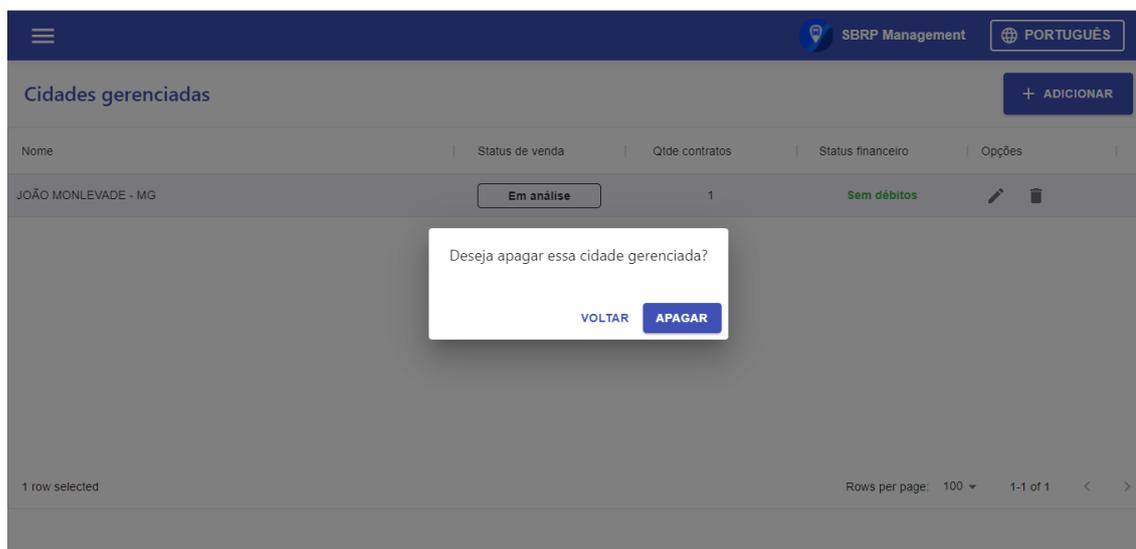
A tabela apresenta cinco colunas, sendo elas:

1. Nome da cidade.
2. Status de venda, sendo eles:
  - **Implantado:** A cidade aderiu ao sistema.
  - **Em análise:** A cidade ainda não deu sua resposta a solicitação ou a reunião de contrato ainda não ocorreu.
  - **Recusado:** A cidade negou a solicitação.
3. Quantidade de arquivos de contrato cadastrados na cidade.
4. Status financeiro da cidade que pode ser:
  - **Débito:** A cidade não quitou seus débitos.
  - **Sem débitos:** A cidade quitou seus débitos.
5. Opções de operações, sendo elas:
  - **Editar:** editar cidade.
  - **Remover:** remover cidade.

## 4.2.2 Remover cidade

A Figura 14 mostra a tela de apagar a cidade cadastrada. O botão *APAGAR* executa a remoção da cidade do banco de dados e da listagem.

Figura 14 – Remover cidade



Fonte: produzido pela autora

### 4.2.3 Adicionar cidade

A Figura 15 mostra a tela de adicionar uma cidade.

Figura 15 – Adicionar cidade

The screenshot shows a web interface for adding a city. At the top, there is a blue header with a menu icon, 'SBRP Management', and a language selector set to 'PORTUGUÊS'. The main content area is titled 'Adicionar cidade'. It features three dropdown menus: 'Adicionar cidade', 'Status de venda' (set to 'Em análise'), and 'Status financeiro' (set to 'Sem débitos'). Below these are two sections: '1. Adicionar gerentes da cidade' and '2. Adicionar contratos'. Section 1 includes a '+' icon and four input fields: 'Nome', 'E-mail', 'Função' (with a dropdown menu showing 'Prefeito'), and 'Telefone'. Section 2 includes a '+' icon, a file upload button 'Escolher arquivo' (with 'Nenhum arquivo selecionado' below it), a 'Módulo' input field, and two date pickers: 'Data inicial' (22/08/2021) and 'Data final' (22/08/2021). At the bottom right are 'VOLTAR' and 'SALVAR' buttons.

Fonte: produzido pela autora

Nessa tela é possível dividir os campos em seções horizontais, sendo elas:

1. Informações sobre a cidade: selecionar o nome da cidade, status de venda e status financeiro.
2. Adicionar gerentes da cidade: adicionar um ou mais gerentes para a cidade informando o nome, *e-mail*, telefone e função. A função pode ser:
  - **Prefeito(a)**
  - **Secretário(a)**
  - **Outros**
3. Adicionar contratos: Caso exista, é possível adicionar o arquivo de contrato, a data de início e término e os módulos contratados da plataforma *SBR-Project*. Os módulos podem ser:
  - **Planejamento**
  - **Monitoramento**

#### 4.2.4 Editar cidade

A Figura 16 mostra a tela de editar uma cidade.

Figura 16 – Editar cidade

The screenshot displays the 'Editar cidade' (Edit city) interface. At the top, there is a blue header with a menu icon, 'SBRP Management', and a language selector set to 'PORTUGUÊS'. The main content area has a title 'Editar cidade' and three dropdown menus: 'Adicionar cidade' (JOÃO MONLEVADE - MG), 'Status de venda' (Em análise), and 'Status financeiro' (Sem débitos). Below these are two sections: '1. Adicionar gerentes da cidade' and '2. Adicionar contratos'. Section 1 contains a form with fields for 'Nome' (Raquel), 'E-mail' (raquel@gmail.com), 'Função' (Prefeito), and 'Telefone' ((31) 99999-9999). Section 2 features a '+ CONTRATO' button with a download icon and a trash icon, followed by a 'Módulo' dropdown with 'Planejamento' and 'Monitoramento' options, and a date range from '22/08/2021' to '28/08/2021'. At the bottom right, there are 'VOLTAR' and 'SALVAR' buttons.

Fonte: produzido pela autora

A tela de edição é bem similar a tela de adição, com a diferença visual no botão de adicionar o contrato, onde o usuário pode baixar o contrato selecionado anteriormente ou excluir o antigo e fazer o upload de um novo arquivo. Essa tela é importante para mudar os status de venda e financeiro e assim facilitar o acompanhamento para o gerente do sistema.

### 4.3 Vendedores

Os vendedores atuam em cidades a fim de apresentar o *SBR-Project* em reuniões com os clientes. Os componentes apresentados nesta Seção são importantes para o gestor porque é possível consultar as informações de contato dos vendedores, visualizar as cidades onde atuam e permite realocá-los caso necessário.

### 4.3.1 Listagem de vendedores

A listagem de vendedores cadastrados no sistema é mostrada na [Figura 17](#).

Figura 17 – Listagem de vendedores

Nome	Telefone	E-mail	Cidades	Opções
Raquel	(31) 99999-9999	raquel@gmail.com	VER CIDADES	
Ana	(31) 98888-8888	ana@gmail.com	VER CIDADES	

1 row selected

Rows per page: 100 1-2 of 2

Fonte: produzido pela autora

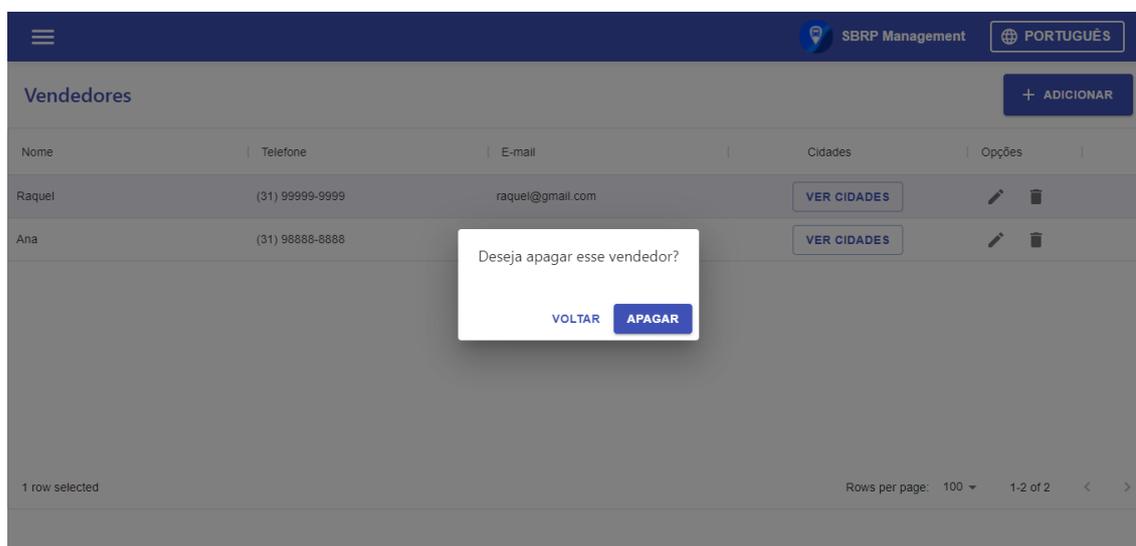
A listagem de vendedores apresenta cinco colunas, são elas:

1. Nome do vendedor.
2. Telefone do vendedor.
3. *E-mail* do vendedor.
4. Cidades: ao clicar no botão exibe um menu informando as cidades onde o vendedor atua.
5. Opções de operações, sendo elas:
  - **Editar:** editar cidade.
  - **Remover:** remover cidade.

### 4.3.2 Remover vendedor

A Figura 18 mostra a tela de apagar o vendedor cadastrado. O botão *APAGAR* executa a remoção do vendedor do banco de dados e da listagem.

Figura 18 – Remover vendedor

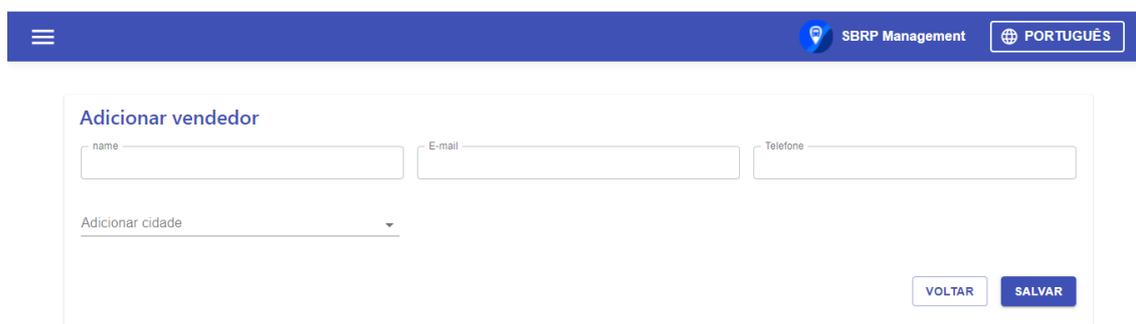


Fonte: produzido pela autora

### 4.3.3 Adicionar vendedor

A Figura 19 mostra a tela de adicionar um vendedor.

Figura 19 – Adicionar vendedor



The screenshot shows the 'Adicionar vendedor' form. At the top, there is a blue header with a hamburger menu icon, the text 'SBRP Management', and a language selector 'PORTUGUÊS'. The form itself has a title 'Adicionar vendedor' and three input fields: 'name', 'E-mail', and 'Telefone'. Below these is a dropdown menu labeled 'Adicionar cidade'. At the bottom right of the form are two buttons: 'VOLTAR' and 'SALVAR'.

Fonte: produzido pela autora

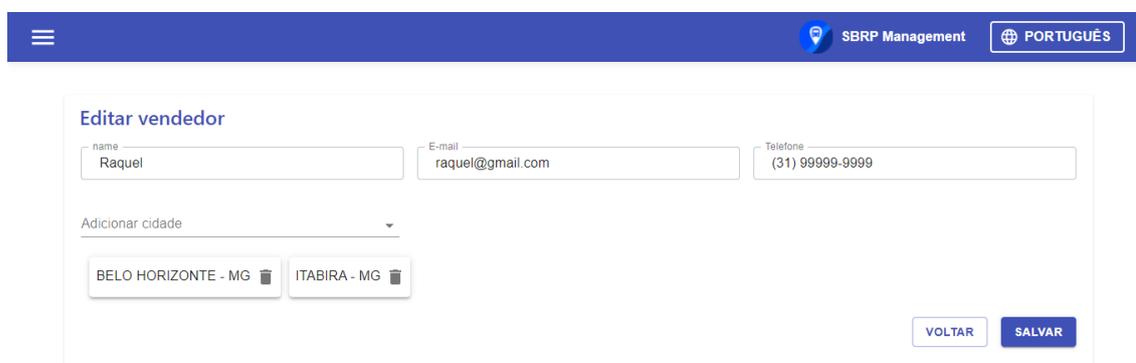
Nessa tela, é possível dividir os campos em seções horizontais, sendo elas:

1. **Informações sobre o vendedor:** cidade, *e-mail* e telefone.
2. **Adicionar cidade:** adicionar uma ou mais cidades onde o vendedor atua.

### 4.3.4 Editar vendedor

A Figura 20 mostra a tela de editar um vendedor.

Figura 20 – Editar vendedor



The screenshot shows the 'Editar vendedor' form. At the top, there is a blue header with a hamburger menu icon, the text 'SBRP Management', and a language selector 'PORTUGUÊS'. The form has a title 'Editar vendedor' and three input fields: 'name' (containing 'Raquel'), 'E-mail' (containing 'raquel@gmail.com'), and 'Telefone' (containing '(31) 99999-9999'). Below these is a dropdown menu labeled 'Adicionar cidade'. Underneath the dropdown are two buttons: 'BELO HORIZONTE - MG' and 'ITABIRA - MG', each with a trash icon. At the bottom right of the form are two buttons: 'VOLTAR' and 'SALVAR'.

Fonte: produzido pela autora

A tela de edição funciona de forma bem similar a tela de adição.

## 4.4 Reuniões

Esta Seção apresenta as telas referente às reuniões realizadas pelos vendedores para apresentar o *SBR-Project* aos clientes. É de grande valia para o gestor, pois permite verificar a data e o resultado de decisão de cada participante. Caso a reunião já tenha sido realizada, é possível ler as observações que os participantes fizeram em casos de recusa, esse retorno é importante a fim de melhorar as estratégias de venda para as próximas negociações.

### 4.4.1 Listagem de reuniões

A listagem de reuniões cadastradas no sistema é mostrada na [Figura 21](#).

Figura 21 – Listagem de reuniões

Data	Cidade	Vendedor	Status	Resultado	Opções
02/11/2021 - 19:36	ITABIRA - MG	RAQUEL	Pendente	-	
02/11/2021 - 19:34	BELO HORIZONTE - MG	ANA	Pendente	-	
02/11/2021 - 19:33	JOÃO MONLEVADE - MG	RAQUEL	Realizado	Aprovado 2  1	

Rows per page: 100 ▾ 1-3 of 3 < >

Fonte: produzido pela autora

A listagem de reuniões para apresenta seis colunas, são elas:

1. Data da reunião.
2. Nome da cidade.
3. Nome do vendedor.
4. Status, sendo eles:
  - **Pendente**
  - **Realizado**

5. Resultado: quando a reunião já ocorreu, é possível visualizar a quantidade de participantes que aceitaram ou recusaram a oferta. Conforme a maioria dos votos é exibido um das opções abaixo:

- **Aprovado**
- **Reprovado**

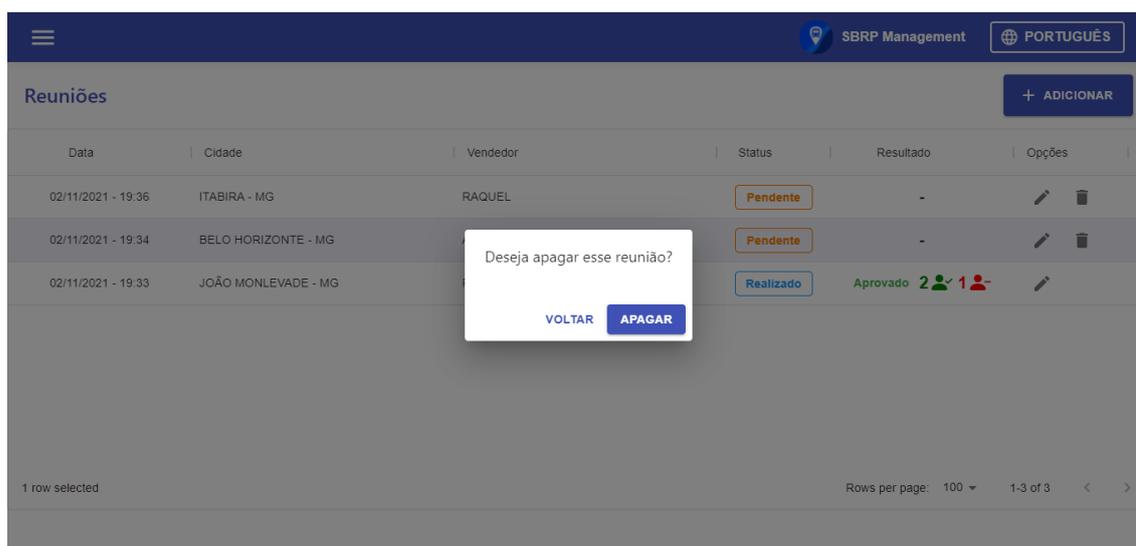
6. Opções de operações, sendo elas:

- **Editar:** editar reunião.
- **Remove:** remover reunião caso ela ainda não tenha ocorrido.

#### 4.4.2 Remover reunião

A [Figura 22](#) mostra a tela de apagar a reunião cadastrada. O botão *APAGAR* executa a remoção da reunião do banco de dados e da listagem.

Figura 22 – Remover reunião



Fonte: produzido pela autora

### 4.4.3 Adicionar reunião

A Figura 23 mostra a tela de adicionar uma reunião.

Figura 23 – Adicionar reunião

A imagem mostra a interface de usuário para adicionar uma reunião. No topo, há uma barra azul com o menu hambúrguer, o logotipo 'SBRP Management' e o idioma 'PORTUGUÊS'. O formulário principal, intitulado 'Adicionar reunião', contém os seguintes elementos:

- Campos de seleção para 'Adicionar cidade' e 'Adicionar vendedor'.
- Um campo de data com o valor '02/11/2021 19:50' e um ícone de calendário.
- Um campo de texto para 'Observação (Opcional)'.
- Uma seção 'Adicionar participante' com um ícone de adição (+), um campo de texto para 'Nome' e um menu suspenso para 'Função' com o valor 'Prefeito' selecionado.
- Dois botões: 'VOLTAR' (branco) e 'SALVAR' (azul).

Fonte: produzido pela autora

Nessa tela, é possível dividir os campos em seções horizontais, sendo elas:

1. Informações sobre a reunião: cidade, vendedor e data que ocorrerá.
2. Observação: breve detalhamento da reunião.
3. Adicionar um ou mais participantes informando nome e função. A função pode ser:
  - **Prefeito(a)**
  - **Secretário(a)**
  - **Outros**

#### 4.4.4 Editar reunião

A Figura 24 mostra a tela de editar uma reunião.

Figura 24 – Editar reunião

**Editar reunião**

Criado em: 02/11/2021 - 19:34

Status: Realizado Resultado: Reprovado

Adicionar cidade: JOÃO MONLEVADE - MG Adicionar vendedor: RAQUEL Data: 02/11/2021 19:33

Observação (Opcional): Reunião para apresentar o produto.

Adicionar participante

Nome: Fernanda	Função: Prefeito	Resultado: Concordou
Nome: Gabriela	Função: Prefeito	Resultado: Concordou
Nome: Bárbara	Função: Prefeito	Resultado: Recusou

Razão da recusa (Opcional): Não teve interesse.

Fonte: produzido pela autora

A tela de edição é importante para mudar o status e registrar o resultado da reunião.

Para cada participante é possível marcar um resultado, sendo eles:

- **Recusou**
- **Se absteve**
- **Concordou**

No caso de recusa, opcionalmente pode ser registrado o motivo dessa decisão.

## 4.5 Considerações finais

Ao final do desenvolvimento do sistema, foram realizados testes manuais para detectar possíveis falhas e melhorias. De acordo com Antunes (2021), teste manual significa testar um aplicativo manualmente por um ser humano. Um especialista em garantia de qualidade (testador) que executa testes manuais garante que um aplicativo esteja funcionando corretamente seguindo as condições descritas nos casos de teste.

Neste Capítulo foram apresentadas as telas desenvolvidas e as operações que podem ser executadas. O próximo Capítulo apresentará a conclusão deste trabalho.

## 5 Conclusão

Este trabalho apresentou os objetivos que levaram o desenvolvimento do projeto e as tecnologias envolvidas. Foram descritos os requisitos e a modelagem do sistema, além da apresentação das telas, funcionalidades e metodologias utilizadas. Por fim foram apresentados os resultados obtidos no desenvolvimento da sistema *web*. A seguir são apresentadas as considerações finais e propostas de trabalhos futuros.

### 5.0.1 Considerações finais

Espera-se que o módulo *SBRP Management* desenvolvido para o sistema *SBR-Project* consiga auxiliar o processo de contratação de serviços para as cidades interessadas. Além de auxiliar na tomada de decisão dos gerentes por meio da visualização dos resultados obtidos nas reuniões de apresentação do sistema ao cliente.

O sistema proposto cumpriu com os objetivos e requisitos especificados para o projeto e obteve resultados satisfatórios. Para que isso ocorresse, foi essencial obter conhecimento sobre práticas para melhorar a experiência do usuário. Entre as práticas estão utilizar um padrão de componentes e cores, mostrar sempre os menus de navegação e desenvolver interfaces sem informações desnecessárias ao usuário.

O software foi desenvolvido utilizando-se tecnologias e padrões de projeto de alto desempenho para o desenvolvimento *web* visando facilitar a escalabilidade e manutenção do sistema tanto no *back-end* quanto no *front-end*. O código da programação apresenta comentários sobre o fluxo de informações a fim de facilitar o entendimento para os próximos desenvolvedores em projetos futuros.

A utilização da biblioteca *React* e da metodologia *Scrum* facilitaram o desenvolvimento ágil da aplicação devido a familiaridade prévia com essas tecnologias e foram fatores primordiais para a diminuição do tempo de desenvolvimento.

### 5.0.2 Trabalhos futuros

Durante o desenvolvimento deste projeto, algumas ideias para novas funcionalidades em projetos futuros foram identificadas. São elas:

- **Integração com serviço de pagamento:** criação de boletos bancários e outros métodos de pagamento existentes.
- **Nota fiscal eletrônica:** Integrar com [API](#) externa para gerar a nota fiscal eletrônica ao realizar o pagamento do plano de serviço.

- 
- **Aplicação móvel:** Desenvolvimento de um aplicativo para uso em celulares e *tablets*.
  - **Exportação de dados:** no formato *.csv* para visualização de dados em uma ferramenta externa.
  - **Teste automatizados:** para uma melhor identificação de possíveis falhas no sistema, tanto no *front-end* quanto no *back-end*.

# Referências

- AGGARWAL, S. Modern web-development using reactjs. *International Journal of Recent Research Aspects*, v. 5, n. 1, p. 133–137, 2018. Citado na página 32.
- ANTUNES, F. *Diferença entre fases de teste, tipos de teste e formas de execução*. [S.l.], 2021. Disponível em: <<https://www.zup.com.br/blog/fases-de-teste-tipos-de-teste>>. Acesso em: 16 dez. 2021. Citado na página 46.
- ANTUNES, M. *API Restful: conceito, princípios e como criar*. [S.l.], 2019. Disponível em: <<https://www.hostgator.com.br/blog/api-restful>>. Acesso em: 23 out. 2021. Citado na página 31.
- ARAÚJO, D. E. M. da C. P. *Metodologia para Cálculo de Custos do Problema de Roteamento de Veículos Escolares*. [S.l.], 2021. Citado na página 16.
- BENHAROSH, J. *What is REST API? in plain English*. [S.l.], 2018. Disponível em: <<https://phpenthusiast.com/blog/what-is-rest-api/>>. Acesso em: 12 jan. 2022. Citado na página 31.
- BRACA J., B. J. P. B. S.-L. D. *A computerized approach to the New York City school bus routing problem*. [S.l.], 1997. Citado na página 19.
- DESROSIERS J., F. J. R. J.-M.-L. G. C. L. *An overview of a school busing system*. [S.l.], 1981. Citado na página 19.
- FEDDERN-BEKCAN, T. Google calendar. *Journal of the Medical Library Association: JMLA*, Medical Library Association, v. 96, n. 4, p. 394, 2008. Citado na página 21.
- Fundo Nacional de Desenvolvimento da Educação. *Guia do Transporte Escolar*. [S.l.], 2019. Disponível em: <<https://www.fnde.gov.br/index.php/centrais-de-conteudos/publicacoes/category/131-transporte-escolar?download=6897:guia-do-transporte-escolar>>. Acesso em: 15 dez. 2021. Citado na página 16.
- GUIMARÃES, E. M. P.; ÉVORA, Y. D. M. Sistema de informação: instrumento para tomada de decisão no exercício da gerência. *Ciência da Informação*, SciELO Brasil, v. 33, p. 72–80, 2004. Citado na página 18.
- Instituto Brasileiro de Geografia e Estatística. [S.l.], 2022. Disponível em: <<https://www.ibge.gov.br/>>. Acesso em: 12 jan. 2022. Citado 2 vezes nas páginas 9 e 10.
- IPEA. *11% dos trabalhadores estiveram em trabalho remoto em 2020 no Brasil*. 3. ed. [S.l.], 2021. Acesso em: 14 nov 2021. Citado na página 22.
- LIMA, L. T. *Ferramenta Web para Rastreamento e Monitoramento de Veículos de Transporte Escolar em Ambientes Reais*. [S.l.], 2021. Citado na página 16.
- MEDEIROS, H. *Introdução ao Padrão MVC*. [S.l.], 2013. Disponível em: <<http://portal.mec.gov.br/ultimas-noticias/384-fnde-1801140772/81021-mec-entrega-180-oni-bus-escolares-a-municipios-paulistas>>. Acesso em: 15 dez. 2021. Citado na página 30.

- MENEZES, D.; OLIVEIRA, S. *MEC entrega 180 ônibus escolares a municípios paulistas*. [S.l.], 2019. Disponível em: <<http://portal.mec.gov.br/ultimas-noticias/384-fnde-1801140772/81021-mec-entrega-180-onibus-escolares-a-municipios-paulistas>>. Acesso em: 15 dez. 2021. Citado na página 16.
- MESSIAS, E. C. *Ferramenta Web para Visualização de Rotas Aplicados ao Problema de Roteamento de Veículos Escolares em Ambientes Reais*. [S.l.], 2018. Citado na página 16.
- NETO, I. da Costa e S. et al. *Um algoritmo heurístico para o problema de roteamento de ônibus escolares*. [S.l.], 2020. Disponível em: <<https://proceedings.science/sbpo-2020/papers/um-algoritmo-heuristico-para-o-problema-de-roteamento-de-onibus-escolares>>. Acesso em: 15 dez. 2021. Citado na página 16.
- NEWTON, R.; THOMAS, W. *Design of school bus routes by computer*. [S.l.], 1969. Citado na página 19.
- OLIVEIRA, H. B. Uma análise jurídica do contrato do modelo de negócio white label. 2021. Citado na página 20.
- PARK, J.; KIM, B.-I. *The school bus routing problem: A review*. [S.l.], 2010. Citado na página 19.
- PRESSMAN, R.; MAXIM, B. *Engenharia de Software-8ª Edição*. [S.l.], 2016. Citado 3 vezes nas páginas 26, 27 e 33.
- SOARES, M. dos S. Metodologias ágeis extreme programming e scrum para o desenvolvimento de software. *Revista Eletrônica de Sistemas de Informação*, v. 3, n. 1, 2004. Citado na página 25.
- WAHYU, E. *React Virtual DOM*. [S.l.], 2020. Disponível em: <<https://medium.com/@ekowahyus/react-virtual-dom-40e0e2781652>>. Acesso em: 12 jan. 2022. Citado na página 32.

# Apêndices

# APÊNDICE A – Telas do sistema SBR-Project

Figura 25 – SBR-Project: Mapa com informações do banco de dados

**Painel de controle** Estado: Minas Gerais Cidade: Belo Horizonte

**Belo Horizonte** Minas Gerais, Brazil

Database Name: city302  
 Database User: marcosshc  
 Database Host: 192.168.2.8  
 Database Port: 5432  
 Ors Key: 5b3ce35978511100c  
 Validated:

TEST CONNECTION REQUEST PASSWORD RESET PASSWORD

EDITAR

Fonte: produzido pela autora

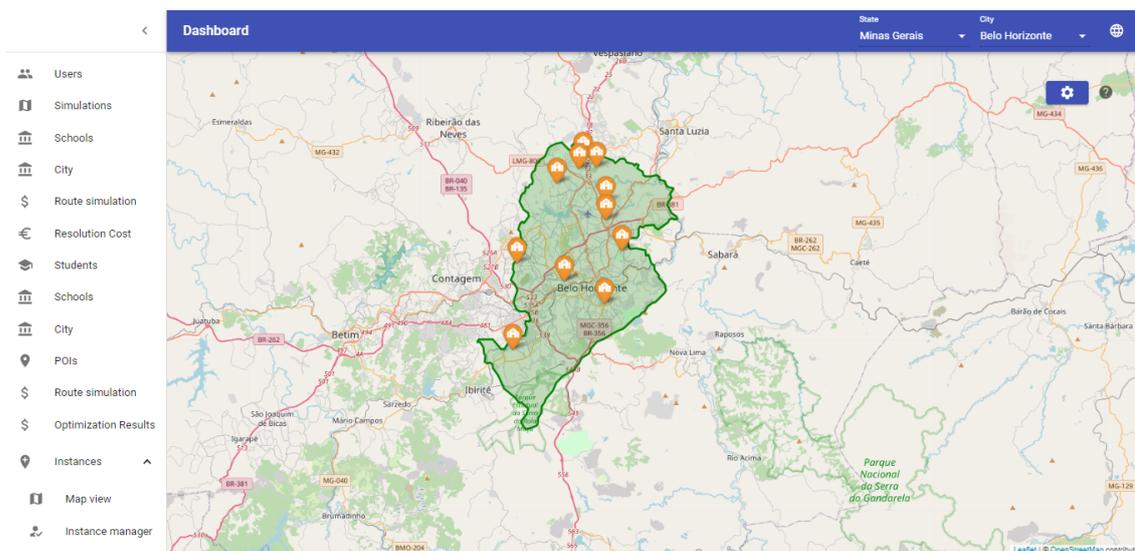
Figura 26 – SBR-Project: Mapa com os marcadores dos pontos de ônibus

**Painel de controle** Estado: Minas Gerais Cidade: Belo Horizonte

Categorias: MGC362

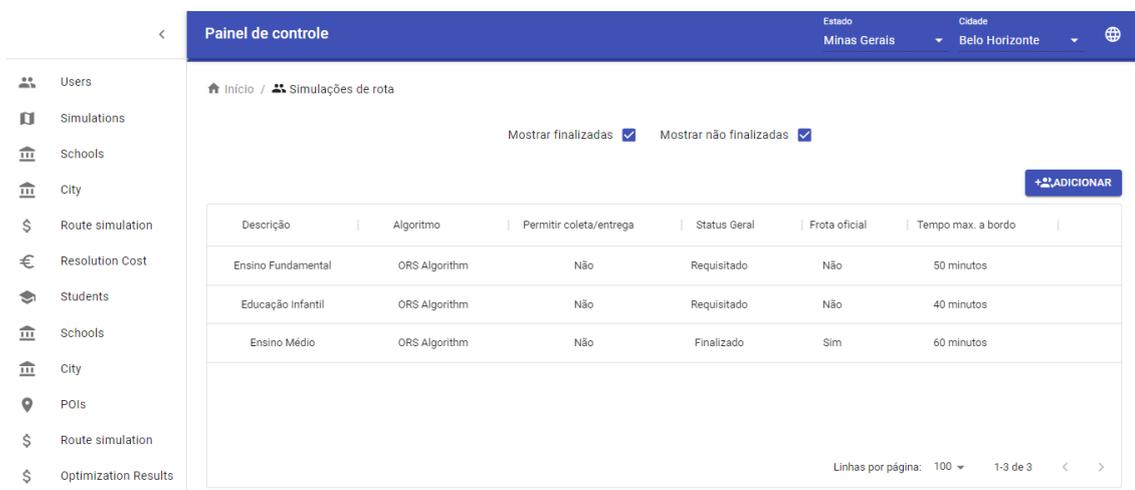
Fonte: produzido pela autora

Figura 27 – SBR-Project: Mapa com os marcadores das residências dos alunos



Fonte: produzido pela autora

Figura 28 – SBR-Project: Informações sobre as rotas



Fonte: produzido pela autora



## **TERMO DE RESPONSABILIDADE**

O texto do Trabalho de Conclusão de Curso intitulado “SBRP Management: Um Sistema de Informação para Gestão de Clientes no Contexto do Transporte Escolar Público” é de minha inteira responsabilidade. Declaro que não há utilização indevida de texto, material fotográfico ou qualquer outro material pertencente a terceiros sem o devido referenciamento ou consentimento dos referidos autores.

João Monlevade, 10 de Janeiro de 2022.

Raquel Martins dos Santos