

**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
CAMPUS CAJAZEIRAS
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS**

**REMÉDIO FÁCIL: UM SISTEMA DE CONTROLE DE DISTRIBUIÇÃO E
DISPONIBILIDADE DE MEDICAMENTOS EM UNIDADES DE SAÚDE**

RUAN MIGUEL COELHO CARDOSO

**Cajazeiras
2024**

RUAN MIGUEL COELHO CARDOSO

**REMÉDIO FÁCIL: UM SISTEMA DE CONTROLE DE DISTRIBUIÇÃO E
DISPONIBILIDADE DE MEDICAMENTOS EM UNIDADES DE SAÚDE**

Trabalho de Conclusão de Curso apresentado junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba - Campus Cajazeiras, como requisito à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Fabio Gomes de Andrade.

**Cajazeiras
2024**

IFPB / Campus Cajazeiras
Coordenação de Biblioteca
Biblioteca Prof. Ribamar da Silva
Catalogação na fonte: Cícero Luciano Félix CRB-15/750

C672r	<p>Coelho, Ruan Miguel Cardoso. Remédio fácil : um sistema de controle de distribuição e disponibilidade de medicamentos em unidades de saúde / Ruan Miguel Cardoso Coelho. – 2024.</p> <p>66f. : il.</p> <p>Trabalho de Conclusão de Curso (Tecnólogo em Análise e Desenvolvimento de Sistemas) - Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, Cajazeiras, 2024.</p> <p>Orientador(a): Prof. Dr. Fabio Gomes de Andrade.</p> <p>1. Desenvolvimento de sistemas. 2. Plataforma remédio Fácil. 3. Controle de estoque de medicamentos. 4. Sistema Único de Saúde. I. Instituto Federal de Educação, Ciência e Tecnologia da Paraíba. II. Título.</p>
-------	--



MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA

RUAN MIGUEL CARDOSO COELHO

**REMÉDIO FÁCIL: UM SISTEMA DE CONTROLE DE DISTRIBUIÇÃO E DISPONIBILIDADE
DE MEDICAMENTOS EM UNIDADES DE SAÚDE**

Trabalho de Conclusão de Curso apresentado junto ao
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas do Instituto Federal de
Educação, Ciência e Tecnologia da Paraíba - Campus
Cajazeiras, como requisito à obtenção do título de
Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador

Prof. Dr. Fabio Gomes de Andrade

Aprovada em: **28 de Fevereiro de 2024.**

Prof. Dr. Fabio Gomes de Andrade - Orientador

Prof. Esp. Afonso Serafim Jacinto - Avaliador

IFPB - Campus Cajazeiras

Prof. Me. Michel da Silva - Avaliador

IFPB - Campus Cajazeiras

Documento assinado eletronicamente por:

- **Fabio Gomes de Andrade**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/03/2024 16:16:29.
- **Afonso Serafim Jacinto**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 01/03/2024 17:54:14.
- **Michel da Silva**, PROFESSOR ENS BASICO TECN TECNOLOGICO, em 04/03/2024 17:47:38.

Este documento foi emitido pelo SUAP em 28/02/2024. Para comprovar sua autenticidade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/autenticar-documento/> e forneça os dados abaixo:

Código 538692
Verificador: 26706c573a
Código de Autenticação:



Rua José Antônio da Silva, 300, Jardim Oásis, CAJAZEIRAS / PB, CEP 58.900-000
<http://ifpb.edu.br> - (83) 3532-4100

RESUMO

O Sistema Único de Saúde (SUS) é hoje um dos maiores e mais complexos sistemas de saúde pública do planeta, que proporciona aos seus usuários uma ampla série de serviços, tais como acompanhamento médico, realização de exames, atendimento de urgência e emergência, vacinação e distribuição de medicamentos. Dentre os serviços disponibilizados, o de distribuição de medicamentos é muitas vezes criticado pela falta de remédios. Esse problema pode ocorrer devido a falhas de comunicação e transparência durante as várias etapas que compõem esse processo. Ao mesmo tempo, ainda não existem ferramentas que informem a população sobre o histórico de disponibilidade de medicamentos nas unidades de distribuição. Com o intuito de solucionar esse problema, este Trabalho de Conclusão de Curso (TCC) propõe a criação de uma plataforma web de escopo nacional chamada Remédio Fácil, que visa aumentar a transparência do serviço de distribuição de medicamentos do SUS, oferecendo informações atualizadas com relação aos medicamentos das unidades de distribuição e um canal direto de comunicação entre a população e pessoas responsáveis pela gestão do serviço.

Palavras-chave: SUS. Medicamento. Transparência. Plataforma Web.

ABSTRACT

The Unified Health System (SUS) is currently one of the largest and most complex public health systems on the planet, which provides its users with a wide range of services, such as medical monitoring, examinations, urgent and emergency care, vaccination and distribution of medicines. Among the services available, medication distribution is often criticized for the lack of medicines. This problem can occur due to failures in communication and transparency during the various steps that make up this process. At the same time, there are still no tools that inform the population about the history of medication availability in distribution units. In order to solve this problem, this Course Completion Work (TCC) proposes the creation of a nationwide web platform called Remédio Fácil, which aims to increase the transparency of the SUS medicine distribution service, offering updated information regarding to medicines from distribution units and a direct channel of communication between the population and people responsible for managing the service.

Keywords: SUS. Medicine. Transparency. Web Platform.

LISTA DE FIGURAS

Figura 1 – Pesquisa no site OpenStreetMap	21
Figura 2 – OpenLayers	22
Figura 3 – Arquitetura da ferramenta	27
Figura 4 – Esquema do banco de dados	30
Figura 5 – Requisição para a rota <i>/medicament/list</i>	33
Figura 6 – Requisição para a rota <i>/pointIsInside</i>	35
Figura 7 – Tela de cadastro de unidade	36
Figura 8 – Modal de pesquisa por endereço	37
Figura 9 – Tela de criação de subordinado	38
Figura 10 – Modo de visualização de cidades ativo	39
Figura 11 – Tela de pesquisa de medicamentos	40
Figura 12 – Informações de uma unidade	41
Figura 13 – Tela de resultado de pesquisa de medicamentos	42
Figura 14 – Opção <i>Medicamentos não encontrados</i>	42
Figura 15 – Informações de uma unidade (Tela de resultado)	43
Figura 16 – Reclamações sobre um medicamento	44
Figura 17 – Modal de criação de reclamação	45
Figura 18 – Tela de histórico do medicamento	46

LISTA DE QUADROS

Quadro 1 – Relação medicament	51
Quadro 2 – Relação distribution_unit	51
Quadro 3 – Relação medicament_distribution_unit	52
Quadro 4 – Relação report	53
Quadro 5 – Relação employee	54
Quadro 6 – Relação attendant	54
Quadro 7 – Relação history	55

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
CATMAT	Catálogo de Materiais
CFM	Conselho Federal de Medicina
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
KML	Keyhole Markup Language
ME	Ministério da Economia
ORM	Object-Relational Mapping
Rename	Relação Nacional de Medicamentos Essenciais
SIASG	Sistema Integrado de Administração de Serviços Gerais
SUS	Sistema Único de Saúde
TCC	Trabalho de Conclusão de Curso
TCESP	Tribunal de Contas do Estado de São Paulo
UBS	Unidades Básicas de Saúde
UTI	Unidade de Terapia Intensiva
WFS	Web Feature Service
WMS	Web Map Service

SUMÁRIO

1	INTRODUÇÃO	15
1.1	MOTIVAÇÃO	15
1.2	TRABALHOS RELACIONADOS	16
1.3	OBJETIVOS	17
1.3.1	Objetivo Geral	17
1.3.2	Objetivos Específicos	17
1.4	METODOLOGIA	18
1.5	ESTRUTURA E ORGANIZAÇÃO DO DOCUMENTO	19
2	FUNDAMENTAÇÃO TEÓRICA	20
2.1	OpenStreetMap e Nominatim	20
2.2	OpenLayers	21
3	A SOLUÇÃO PROPOSTA	23
3.1	ANÁLISE E PROJETO	23
3.1.1	Atores da ferramenta	23
3.1.2	Requisitos Funcionais	24
3.1.3	Projeto arquitetural	27
3.1.4	Banco de dados espacial	28
3.1.5	Banco de dados de edição	29
3.2	IMPLEMENTAÇÃO	32
3.2.1	Módulo de Edição	32
3.2.2	Módulo espacial	33
3.2.3	Módulo de visão	35
4	CONSIDERAÇÕES FINAIS	47
4.1	CONCLUSÃO	47
4.2	TRABALHOS FUTUROS	47

REFERÊNCIAS	49
APÊNDICE A – DICIONÁRIO DE DADOS	51
APÊNDICE B – ENDPOINTS DA API DE EDIÇÃO	56
APÊNDICE C – ENDPOINTS DA API ESPACIAL	67

1 INTRODUÇÃO

De acordo com o site da Fundação Fernando Henrique Cardoso, antes da constituição federal de 1988, “a saúde era vinculada ao *Instituto de Assistência Médica da Previdência Social* (Inamps), sendo os serviços assegurados apenas a quem contribuisse” (FUNDAÇÃO FHC, 2023), negando assim atendimento à uma parte da população brasileira. Essa situação mudou apenas com a promulgação da constituição federal de 1988, mais especificamente com seu artigo 196, que afirma que:

A saúde é direito de todos e dever do Estado, garantido mediante políticas sociais e econômicas que visem à redução do risco de doença e de outros agravos e ao acesso universal e igualitário às ações e serviços para sua promoção, proteção e recuperação (BRASIL, 1988, cap II, art. 196).

Ainda segundo a Fundação FHC, esse artigo da constituição serviu como uma das bases para a criação do SUS, que hoje atua em vários serviços, como consultas com médicos e realização de cirurgias. Atualmente, o SUS “é um dos maiores e mais complexos sistemas de saúde pública do mundo” (GOV.BR, s.d.), mas, ainda assim, possui problemas que prejudicam a assistência à população, como o grande tempo de espera pelo atendimento e a dificuldade para a obtenção de vagas em leitos de Unidade de Terapia Intensiva (UTI). Por exemplo, segundo uma pesquisa realizada pelo Datafolha (2018), sob pedido do Conselho Federal de Medicina (CFM), cerca de 55% dos entrevistados avaliaram a saúde no Brasil como péssima ou ruim. Mas, apesar do descontentamento por parte da população, os serviços do SUS ainda são fortemente utilizados. Conforme um levantamento da mesma pesquisa, cerca de 97% dos entrevistados chegaram a utilizar algum dos serviços do SUS em um intervalo de dois anos, o que destaca a sua grande presença e importância na vida do brasileiro. Dentre os serviços disponibilizados pelo SUS, um dos mais utilizados é a distribuição de remédios gratuitos, tendo sido utilizado por cerca de 62% dos entrevistados.

1.1 MOTIVAÇÃO

Atualmente, o serviço de distribuição de remédios gratuitos sofre com a falta de remédios em algumas unidades distribuidoras, como pode ser observado no relatório da 1ª Fiscalização Ordenada de 2023 realizada pelo Tribunal de Contas do Estado de São Paulo (TCESP) (2023), que mostra que 46,87% das unidades fiscalizadas apresentaram falta de medicamentos que elas deveriam dispensar. Devido a essa falta

de medicamentos, há pessoas que precisam se locomover por grandes distâncias para obtê-los em outras unidades, assim como pessoas que não os conseguem através do SUS, e que, portanto, precisam obtê-los em farmácias privadas ou interromper seu uso, sendo que essa última opção pode trazer graves riscos à sua saúde.

Apesar dos evidentes problemas no serviço de distribuição de medicamentos pelo SUS, ainda não há dados precisos sobre a quantidade de cidadãos que não conseguem obter os medicamentos de que precisam, mesmo tendo o direito de recebê-los. Além disso, a falta de informação sobre os medicamentos disponíveis em cada unidade de distribuição, aliada à má distribuição dos medicamentos entre as unidades, faz com que pessoas tenham que fazer um grande deslocamento até as unidades de distribuição e, mesmo assim, não consigam receber os seus medicamentos. Esse problema é ainda maior em pacientes que possuem dificuldade de locomoção, tais como idosos ou pessoas com mobilidade reduzida.

Com o objetivo de amenizar essas dificuldades, este Trabalho de Conclusão de Curso (TCC) desenvolveu o *Remédio Fácil*, uma ferramenta que fornece à população informações atualizadas sobre a disponibilidade de medicamentos em unidades de distribuição do SUS. Além disso, a ferramenta faz o controle do histórico dos remédios de cada unidade.

1.2 TRABALHOS RELACIONADOS

A ferramenta *Remédio Fácil*, proposta neste TCC, possui similaridades com outras ferramentas já desenvolvidas. Dessas ferramentas, três foram selecionadas para serem comparadas com a *Remédio Fácil*: a *Aqui Tem Remédio*¹, a *Meu Remédio Sorocaba*² e o site da prefeitura de Porto Alegre³.

Segundo seu manual de instruções (Secretaria Municipal da Saúde de São Paulo, s.d.), a ferramenta *Aqui Tem Remédio* está disponível como site e aplicativo de celular. Essa ferramenta precisa que seu usuário informe um grupo de medicamentos e uma localização e, então, realize a busca. Como resultado, são retornadas as farmácias da rede de saúde do município de São Paulo que possuem ao menos um dos medicamentos procurados. Essa ferramenta se distingue da proposta neste TCC por não disponibilizar a visualização do histórico dos medicamentos e por sua abrangência, que foca apenas o município de São Paulo, enquanto a *Remédio Fácil* cobre todo o país.

¹ https://play.google.com/store/apps/details?id=br.com.insix.aquitemremedio&hl=pt_BR

² <http://winest-pms.humanconcierge.com.br/posicaoestoquesimples>

³ <https://prefeitura.poa.br>

Ao utilizar o site *Meu Remédio Sorocaba*, percebeu-se que ele permite que seu usuário pesquise pelo nome de um medicamento no estoque de uma ou de todas as Unidades Básicas de Saúde (UBS) cadastradas. A ferramenta, então, mostra o código e a descrição dos medicamentos que corresponderam ao nome buscado, as UBSs em que esses medicamentos estão disponíveis e a quantidade desses medicamentos em cada uma dessas UBSs. Essa ferramenta difere da *Remédio Fácil*, em primeiro lugar, por permitir a busca de apenas um medicamento por vez, enquanto a *Remédio Fácil* permite buscar por um grupo com um ou mais medicamentos. Em segundo lugar, a *Meu Remédio Sorocaba* só é capaz de realizar pesquisas em uma ou em todas as unidades cadastradas, enquanto a *Remédio Fácil* retorna apenas as informações das unidades localizadas dentro do raio especificado na busca. E, por último, a *Meu Remédio Sorocaba* não disponibiliza a visualização do histórico dos medicamentos e cobre apenas a cidade de Sorocaba, diferente da *Remédio Fácil*.

O site da prefeitura de Porto Alegre também oferece o serviço *Onde Está Seu Medicamento*⁴. Ao fazer uso dele, percebeu-se que ele permite que seu usuário selecione um dos medicamentos cadastrados no sistema e pesquise por sua quantidade nas farmácias do município. Entre as diferenças desse serviço para o serviço de busca de medicamentos da *Remédio Fácil*, destaca-se que o primeiro possibilita a busca de apenas um medicamento por vez, enquanto o segundo permite a busca por um grupo de medicamentos. Outra diferença é que o primeiro permite que o usuário filtre os nomes das farmácias retornadas no resultado da busca, enquanto o segundo filtra as unidades retornadas com base na distância delas para a posição do usuário. E, por último, o primeiro cobre apenas Porto Alegre, enquanto o segundo cobre todo o Brasil.

1.3 OBJETIVOS

As próximas seções descrevem o objetivo geral e os objetivos específicos deste trabalho.

1.3.1 Objetivo Geral

Este TCC tem como objetivo geral a implementação de uma plataforma web para melhorar a qualidade e a transparência do serviço de distribuição de medicamentos gratuitos pelo SUS.

1.3.2 Objetivos Específicos

O trabalho também tem os seguintes objetivos específicos:

⁴ <https://prefeitura.poa.br/carta-de-servicos/acesso-medicamentos-e-insumos>

- disponibilizar para o cidadão informações atualizadas sobre os remédios disponíveis nas unidades de distribuição sem a necessidade de ir até elas ou de entrar em contato com seus funcionários;
- mostrar para o cidadão o histórico de disponibilidade dos remédios de cada unidade de distribuição;
- unificar, em apenas uma ferramenta, a gestão da distribuição de medicamentos na rede pública;
- permitir que o cidadão conteste a veracidade das informações dispostas na ferramenta;
- possibilitar a criação de estratégias para a melhoria do serviço de distribuição de medicamentos pelo SUS, com base nos dados salvos no histórico de disponibilidade.

1.4 METODOLOGIA

Nesta seção são abordadas as subdivisões do processo de concepção deste TCC, doravante chamadas de atividades, que compreendem desde a sua idealização até sua implementação e documentação. As atividades definidas foram as seguintes:

- **Pesquisa por ferramentas similares à proposta neste TCC (A1):** esta atividade englobou a busca por ferramentas que abordam o mesmo problema e realizam atividades similares à ferramenta proposta neste TCC, em busca de verificar a sua unicidade;
- **Levantamento de requisitos (A2):** nesta atividade foi definida a restrição da abrangência esperada da ferramenta, com a definição dos atores e dos requisitos funcionais;
- **Definição da estrutura do banco de dados (A3):** nesta atividade foi feita a construção, com base nos resultados da atividade anterior, do esquema lógico do banco de dados utilizado pela ferramenta;
- **Definição da arquitetura e infraestrutura da ferramenta (A4):** esta atividade abarcou a delimitação das tecnologias utilizadas na ferramenta e as decisões acerca de sua infraestrutura;
- **Implementação da ferramenta (A5):** esta atividade foi constituída pela implementação dos três módulos que compõem a arquitetura da ferramenta, descrita na seção 3.1.3;

- **Elaboração do documento (A6):** nesta atividade foi realizada a escrita deste documento.

1.5 ESTRUTURA E ORGANIZAÇÃO DO DOCUMENTO

O restante deste documento está organizado em três capítulos:

- O Capítulo 2 retrata a fundamentação teórica, apresentando as tecnologias utilizadas para o desenvolvimento deste trabalho;
- O Capítulo 3 descreve a ferramenta proposta neste trabalho e seus aspectos, apresentando informações como os requisitos funcionais levantados, a arquitetura que foi utilizada para a implementação e o esquema lógico usado para a implementação do banco de dados;
- O Capítulo 4 compreende as considerações finais deste TCC, abordando a conclusão deste trabalho e os possíveis trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo descreve as principais tecnologias utilizadas na produção deste TCC. Inicialmente são descritas as tecnologias *OpenStreetMap* (HAKLAY; WEBER, 2008) e *Nominatim*¹, ambas relacionadas ao uso de dados espaciais. Em seguida é descrito o *OpenLayers* (HAZZARD, 2011), que foi utilizado para descrever dados espaciais em mapas.

2.1 OPENSTREETMAP E NOMINATIM

O *OpenStreetMap* é um projeto de dados abertos de mapeamento colaborativo no qual voluntários contribuem com as informações para a construção e atualização dos mapas. Esses voluntários não são limitados apenas a pessoas físicas, podendo ser também companhias ou agências governamentais (BENNETT, 2010).

Como é possível ver em sua documentação², os dados disponibilizados por este projeto podem ser utilizados no formato de *tile layers*, forma de renderização que utiliza conjuntos de imagens para compor o mapa, o que é útil para situações em que o objetivo principal é mostrar o mapa em uma interface gráfica. Além disso, ainda de acordo com sua documentação³, os dados do projeto também podem ser baixados para que possam ser importados para um banco de dados.

O *Nominatim*, por sua vez, segundo sua documentação⁴, é uma ferramenta de código aberto que utiliza os dados do *OpenStreetMap* para geocodificação (conversão de endereços em coordenadas geográficas) e geocodificação reversa (conversão de coordenadas geográficas em endereços). Ela inclusive é utilizada no site do próprio *OpenStreetMap*, como pode ser visto na Figura 1, que mostra o resultado da pesquisa pelo texto “Cajazeiras” no site do *OpenStreetMap*.

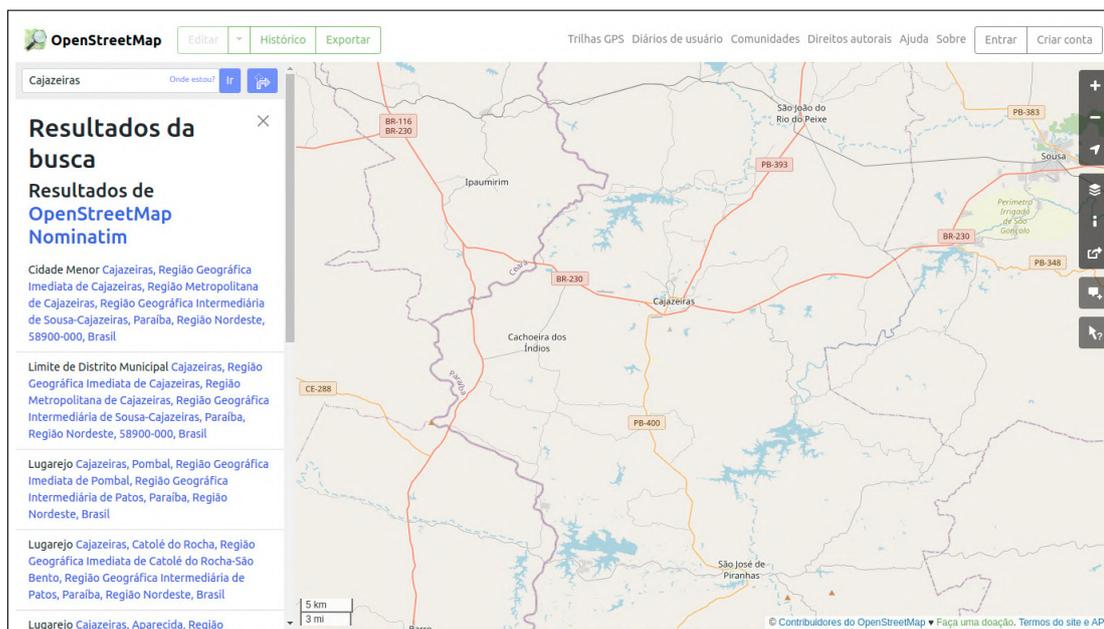
¹ <https://nominatim.org>

² https://wiki.openstreetmap.org/wiki/Featured_tile_layers [Acesso em 10 de mar. de 2024].

³ https://wiki.openstreetmap.org/wiki/Pt:Downloading_data [Acesso em 10 de mar. de 2024].

⁴ <https://nominatim.org/release-docs/latest> [Acesso em 10 de mar. de 2024].

Figura 1 – Pesquisa no site OpenStreetMap



Fonte: Site <https://www.openstreetmap.org>

2.2 OPENLAYERS

O *OpenLayers* é uma biblioteca de código aberto para a renderização de mapas em ambientes web. Ele reúne diversos recursos para a apresentação de dados espaciais, tanto em forma de imagens quanto de vetores, separando esses dados em camadas sobrepostas (HAZZARD, 2011).

Essa ferramenta também dá suporte a vários formatos como, por exemplo, GeoJSON (BUTLER et al., 2016) e KML (Keyhole Markup Language) (WERNECKE, 2008), e ainda fornece construtores para requisições e ainda fornece construtores para requisições *Web Map Service* (WMS) (HAZZARD, 2011) e *Web Feature Service* (WFS) (HAZZARD, 2011), que são serviços *web* usados para requisição de dados espaciais.

Além disso, o *OpenLayers* permite a interação do usuário com os mapas e a customização dessa interação por parte do programador. É possível, por exemplo, permitir ao usuário desenhar formas geométricas sobre os mapas ou alterar a aparência dos mapas com base nas ações do mouse ou teclado, além de várias outras possíveis customizações.

Na Figura 2 é mostrado um mapa gerado utilizando a ferramenta *OpenLayers*.

As imagens que compõem esse mapa foram fornecidas pelo *OpenStreetMap*, utilizando uma integração já presente na ferramenta *OpenLayers*. A visualização em questão foi propositalmente focada no território brasileiro.

Figura 2 – OpenLayers



Fonte: Elaborado pelo autor

3 A SOLUÇÃO PROPOSTA

Este capítulo descreve o processo de construção do *Remédio Fácil*, que é a ferramenta desenvolvida neste TCC, cujo objetivo consiste em melhorar a qualidade e a transparência do serviço de distribuição de medicamentos gratuitos pelo SUS. Inicialmente, o capítulo aborda os artefatos produzidos nas etapas de análise e projeto da ferramenta, sendo eles os atores, os requisitos funcionais, o projeto arquitetural e o esquema lógico do banco de dados. Em seguida, o capítulo aborda a infraestrutura da ferramenta, e a descrição de toda a sua implementação.

3.1 ANÁLISE E PROJETO

Esta seção apresenta os artefatos produzidos durante a etapa de análise e projeto da ferramenta.

3.1.1 Atores da ferramenta

A ferramenta foi projetada para ser utilizada por quatro tipos de atores, tendo em vista o objetivo de mostrar informações atualizadas para os cidadãos, e a necessidade de outros atores para alimentarem a ferramenta com essas informações. São eles:

- **cidadão:** representa qualquer pessoa que queira saber a quantidade, atual ou em algum momento no passado, de um ou mais medicamentos nas unidades de distribuição;
- **gerente regional:** responsável por controlar as unidades de distribuição de medicamentos sobre as quais têm posse e os seus respectivos atendentes. Ele também pode controlar os gerentes dos quais é superior direto;
- **administrador:** representa um nível maior de gerente, com autonomia sobre todo o território nacional. É o único gerente cadastrado previamente no sistema e, portanto, superior, direto ou indireto, de todos os outros gerentes. Esse também é o único ator que pode cadastrar novos medicamentos;
- **atendente:** ator que representa os atendentes das unidades de distribuição de medicamentos, sendo o responsável por atualizar a quantidade dos medicamentos e gerenciar as reclamações de sua unidade de distribuição.

3.1.2 Requisitos Funcionais

Os requisitos funcionais para esta ferramenta foram definidos com o objetivo de acatar os objetivos gerais e específicos definidos no capítulo 1.3, garantindo que a mesma auxilie os usuários na busca de medicamentos e visualização de seus históricos de disponibilidade, além de possuir um esquema de hierarquia no seu lado administrativo. A ferramenta deve satisfazer os seguintes requisitos funcionais:

- **autenticar gerentes e atendentes (RF1):** a ferramenta deve permitir que o administrador e os gerentes se autenticuem no sistema utilizando seu login e senha. Os atendentes também devem ser capazes de logar no sistema porém, utilizando seu login, senha e especificando a unidade a que pertencem;
- **controlar os medicamentos distribuídos gratuitamente pelo SUS (RF2):** a ferramenta deve possuir uma lista com o nome, forma e composição ou concentração de todos os medicamentos disponibilizados gratuitamente pelo SUS. E o administrador deve ser capaz de adicionar novos medicamentos a essa lista;
- **criar gerentes (RF2):** caso um gerente tenha permissão para tal, a ferramenta deve permitir que ele adicione outros gerentes de região, que ficam automaticamente sob o seu controle e passam a ser considerados seus subordinados diretos. Para cada gerente, devem ser armazenados seu nome, login, senha, nível de permissão e área de autoridade. O nível de permissão mínimo de um gerente deve ser o de controlar unidades de distribuição, também podendo receber permissão para controlar outros gerentes de região. A área de autoridade de um gerente deve ser sempre um subconjunto da área de autoridade de seu superior direto;
- **atualizar informações de gerente (RF4):** a ferramenta deve permitir que um gerente altere seu nome e senha;
- **atualizar área de autoridade de subordinados diretos (RF5):** a ferramenta deve permitir que um gerente altere a área de autoridade de seus subordinados diretos. Porém, a nova área deve ser sempre um subconjunto da área do próprio gerente. Quando um gerente remover uma parte da área de autoridade de um subordinado, ele deve receber as unidades de distribuição desse subordinado que estiverem localizadas dentro da área removida. Além disso, ele também pode receber subordinados do gerente que perdeu área, pois este não pode manter subordinados com autoridade sob uma área que ele mesmo não possui. Como exemplo, considere-se a existência de três gerentes: A, B e C, sendo que o gerente A tem autoridade sobre as cidades de Cajazeiras, Sousa e Ipaumirim, o gerente B sendo um subordinado direto de A, com autoridade sobre Cajazeiras

e Sousa. E, por último, o gerente C sendo um subordinado direto de B, com autoridade sobre Cajazeiras. Caso o gerente A remova do gerente B a autoridade sobre a cidade de Cajazeiras, o gerente C deixa de ser subordinado direto do gerente B e passa a ser subordinado direto de A;

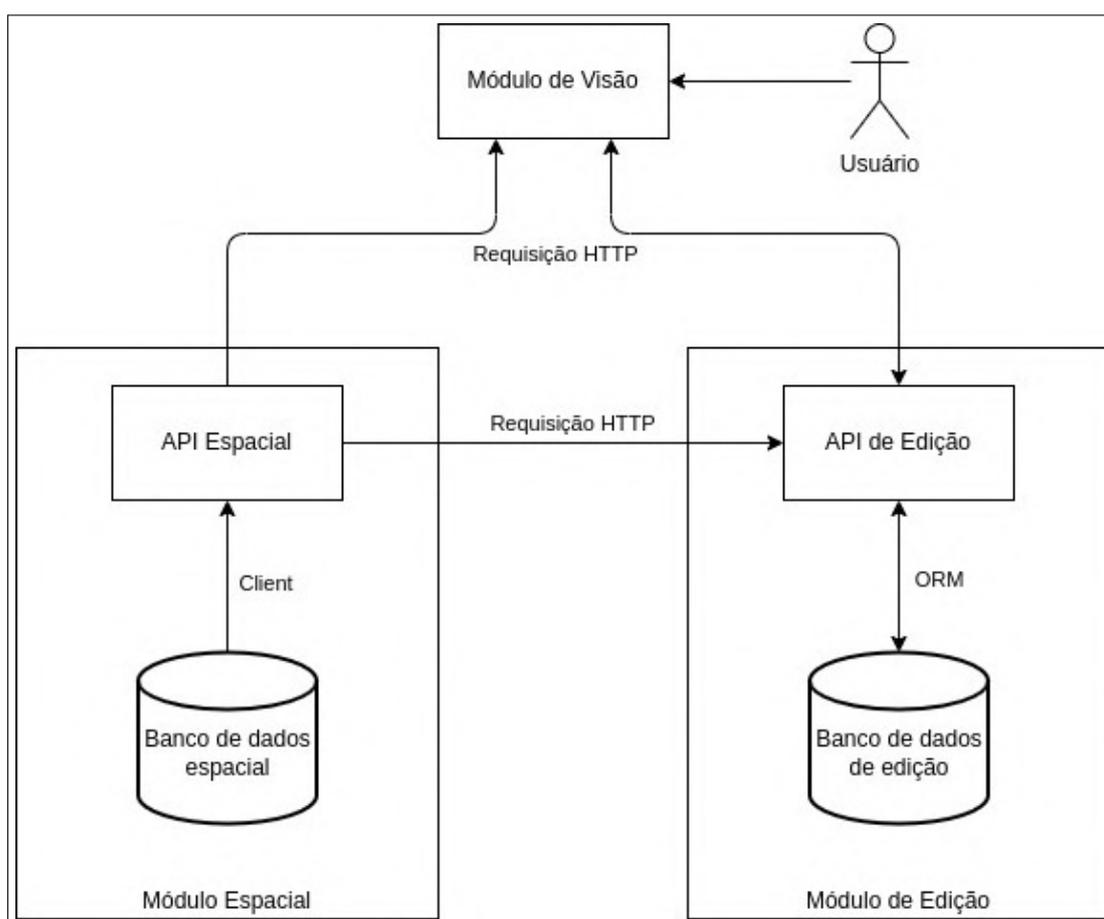
- **atualizar o nível de permissão de subordinados diretos (RF6):** a ferramenta deve permitir que um gerente altere o nível de permissão de seus subordinados diretos. Se um gerente A der a um subordinado B permissão para controlar outros gerentes, o gerente A não sofre mudança alguma mas, se o gerente A remover essa permissão do subordinado B, A deve receber todos os subordinados diretos de B;
- **remover subordinados diretos (RF7):** a ferramenta deve permitir que um gerente exclua a conta de seus subordinados diretos. Quando ele fizer isso, deve receber todas as unidades de distribuição e subordinados diretos do gerente cuja conta foi excluída;
- **transferir subordinados diretos (RF8):** a ferramenta deve permitir que um gerente transfira seus subordinados diretos para outro de seus subordinados diretos. Para isso, o subordinado que receberá os outros deve ter permissão para controlar gerentes. Além disso, também deve ser possível que um gerente tome para si os subordinados diretos de seus subordinados diretos;
- **criar unidades de distribuição (RF9):** a ferramenta deve permitir que um gerente crie unidades de distribuições dentro da área em que este tenha autoridade e, ao ser criada, esta unidade deve ficar sob posse de seu criador. Para cada unidade, devem ser mantidos os seguintes dados: nome, localização, horário de funcionamento e o estoque de cada medicamento distribuído pela unidade;
- **gerenciar unidades de distribuição (RF10):** um gerente que tenha posse de uma unidade de distribuição deve ser capaz de excluí-la e alterar seu nome, localização e horário de funcionamento. Além disso, ele também deve poder transferir a posse da unidade para um de subordinados diretos, assim como de tomar para si as unidades de seus subordinados diretos;
- **gerenciar atendentes (RF11):** a ferramenta deve permitir que um gerente crie e remova atendentes das unidades de distribuição que possui. Para cada atendente, devem ser armazenados seu nome, login e senha;
- **atualizar informações de atendente (RF12):** a ferramenta deve permitir que um atendente altere seu nome e senha;

- **alterar quantidade dos medicamentos (RF13):** a ferramenta deve permitir que os atendentes das unidades de distribuição possam alterar a quantidade dos medicamentos de sua unidade de distribuição;
- **pesquisar por conjuntos de medicamentos (RF14):** a ferramenta deve permitir que os cidadãos façam buscas por grupos de medicamentos contidos na lista resultante do requisito RF2. Essas buscas devem usar como base a localização atual do usuário e um raio de busca definido por ele. Ao realizar essa consulta, a ferramenta deve mostrar todas as unidades dentro do raio da busca e, para cada uma dessas unidades, mostrar quais dos medicamentos procurados estão disponíveis no momento. Para cada medicamento disponível deve ser mostrada sua quantidade e, caso esse valor seja maior do que cinco, deve informar apenas que ela é maior que este valor, não informando o número real. Esse ocultamento do número real de unidades é adotado para evitar que esse dado seja utilizado no planejamento de roubos às unidades;
- **controlar reclamações de inconsistência (RF15):** a ferramenta deve permitir que os usuários informem erros nas informações dos medicamentos em uma unidade de distribuição. Para isso, o usuário deve informar seu email (para o qual pode ser enviada uma resposta), o medicamento cuja informação está incorreta, se a reclamação é pública ou não, e o conteúdo da sua reclamação. Além disso, deve ser armazenado o horário em que a reclamação foi feita. Ainda com relação a esse requisito, caso um medicamento tenha recebido uma ou mais reclamações públicas nos últimos 30 dias, buscas por ele nessa mesma unidade devem mostrar aos usuários o número total dessas reclamações e seu conteúdo;
- **responder reclamações de inconsistência (RF16):** a ferramenta deve permitir que atendentes respondam reclamações feitas sobre os medicamentos de sua unidade de distribuição. Essas respostas devem ser enviadas para o email atrelado à reclamação e, em caso de reclamações públicas, a resposta deve aparecer junto à reclamação quando esta é mostrada em alguma pesquisa;
- **controlar o histórico de disponibilidade (RF17):** cada unidade de distribuição deve ter o próprio histórico de disponibilidade para cada um de seus medicamentos. O histórico deve guardar todos os dias a quantidade de cada medicamento e o número de reclamações feitas sobre este no dia em questão. O histórico de um medicamento porém, só deve ficar disponível para visualização 30 dias após sua data de referência, para evitar que esse dado seja utilizado no planejamento de roubos às unidades, complementando a ocultação do número de medicamentos do requisito RF14.

3.1.3 Projeto arquitetural

Esta seção apresenta a arquitetura da ferramenta proposta neste TCC. A arquitetura usada para a sua implementação é mostrada na Figura 3. Nela, observa-se que a ferramenta foi implementada como uma plataforma web, para que o usuário não precise instalar nada para utilizá-la, que se divide em três módulos principais: visão, edição e espacial.

Figura 3 – Arquitetura da ferramenta



Fonte: Elaborado pelo autor

Como mostrado na Figura 3, o *módulo de visão* é o único módulo acessado diretamente pelos usuários finais. Ele é responsável por possibilitar que o cliente utilize a ferramenta mesmo sem possuir qualquer conhecimento em programação, funcionando como um intermediário entre o usuário e os outros módulos, com os quais se comunica por meio de requisições HTTP (*Hypertext Transfer Protocol*), que é um protocolo de transferência de dados pela internet.

O *módulo de edição* é encarregado da manutenção e leitura do estado do sistema. Para tanto, ele possui os próprios banco de dados, o banco de dados espacial, e uma API web, a API de edição. O banco de dados de edição é um banco de dados relacional, cujo propósito é guardar os dados necessários para o funcionamento de seu módulo. Já a API de edição é responsável por padronizar e restringir a troca de informações entre seu módulo e os outros dois módulos.

O ORM (*Object-Relational Mapping*), que estabelece a comunicação entre o banco de dados de edição e a sua correspondente API, por sua vez, é uma ferramenta que simplifica a comunicação entre um banco de dados relacional e uma linguagem de programação com suporte à programação orientada a objetos.

O *módulo espacial*, por sua vez, é incumbido de disponibilizar as informações de geolocalização das áreas dos estados e cidades brasileiras aos outros dois módulos. Para tal, ele faz uso de seu banco de dados, o banco de dados espacial, e sua API, a API espacial. O banco de dados espacial é o responsável por armazenar as informações espaciais necessárias para esse módulo executar a sua tarefa. Já a API espacial é a responsável por se comunicar com os outros dois módulos por meio de requisições HTTP e tratar os dados armazenados no banco de dados espacial de modo a suprir as demandas dos mesmos. A comunicação entre o banco de dados e API espacial se dá por intermédio de uma interface de cliente, ferramenta responsável por padronizar a comunicação entre a linguagem de programação e o banco de dados.

Um detalhe importante é que a comunicação entre o *módulo espacial* e os outros dois módulos é unilateral. Assim, ele em momento algum faz requisições, apenas as recebe e responde. Além disso, esse módulo não altera as informações armazenadas em seu banco de dados, apenas as recupera.

3.1.4 Banco de dados espacial

O *banco de dados espacial* é responsável pelo armazenamento dos dados necessários para o funcionamento da API espacial. Ele possui apenas duas tabelas: *planet_osm_polygon* e *possession*. A primeira representa as cidades e estados brasileiros, armazenando os polígonos referentes a suas áreas, além de outras informações como seus nomes e, no caso dos estados, suas siglas. Essa tabela foi importada do OSM e, salvo a remoção de tuplas desnecessárias para o escopo do projeto, seu conteúdo não foi alterado, mantendo, portanto, sua estrutura original.

A importação da tabela *planet_osm_polygon* foi feita a partir de dados obtidos do serviço de extração de dados fornecido pela *Geofabrik*, que é uma companhia que

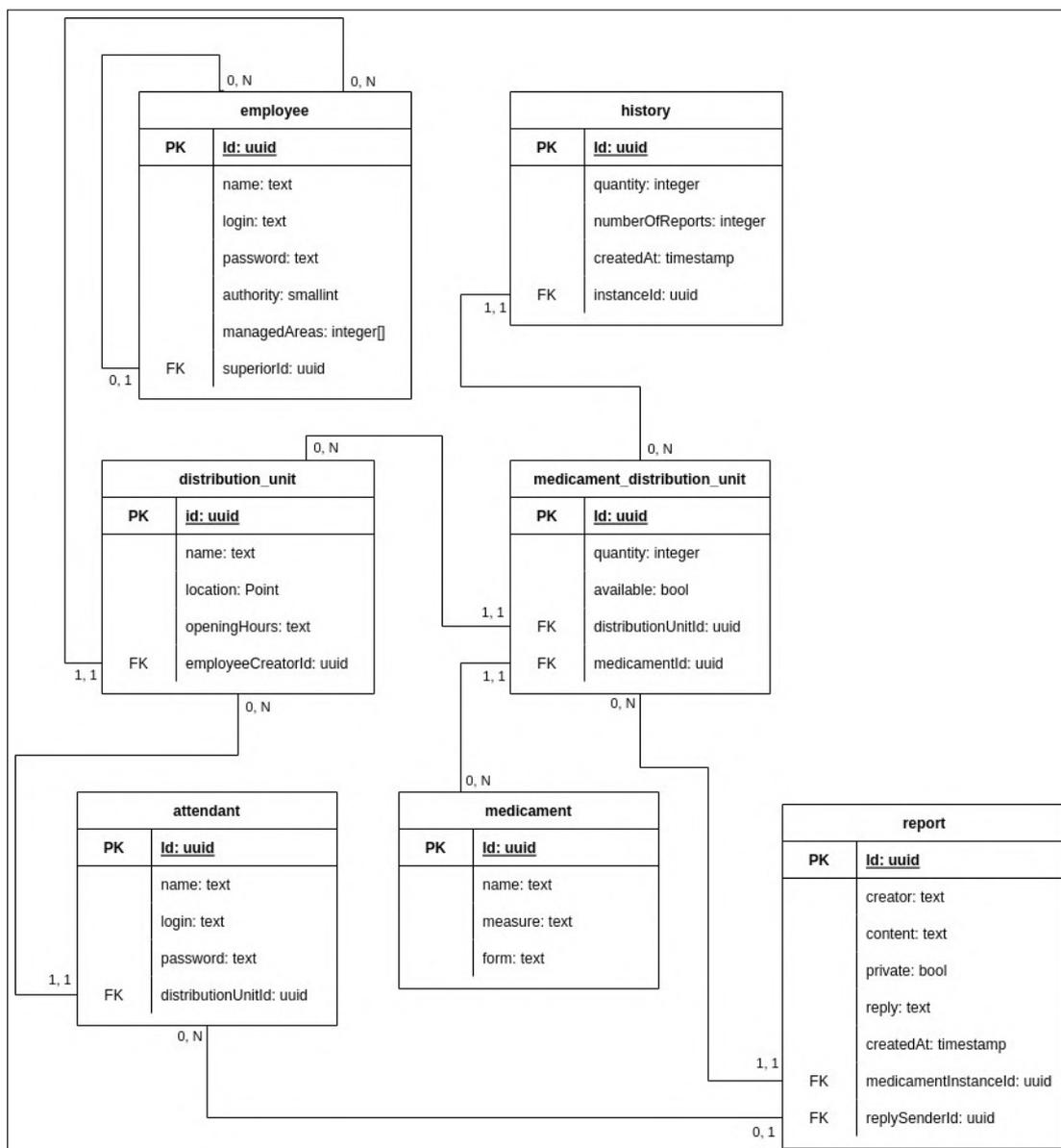
extrai, seleciona e processa geodados gratuitos, além de oferecer serviços de consultoria e treinamento e criar soluções web que envolvem o uso de mapas. Foi baixado um arquivo no formato PBF que continha o subconjunto dos dados do *OpenStreetMap* referentes ao Brasil. Esse arquivo foi importado no banco de dados com o auxílio do programa *osm2pgsql*, e, por fim, os dados importados foram filtrados buscando manter apenas os limites dos estados e cidades brasileiros.

A tabela *possession* foi gerada a partir dos dados da tabela *planet_osm_polygon* com o objetivo de agilizar a identificação de quais cidades compõem cada estado. Essa tabela possui apenas duas colunas: *city* e *uf*, que guardam, respectivamente, o identificador de uma cidade, e o identificador do estado que esta cidade faz parte.

3.1.5 Banco de dados de edição

O *banco de dados de edição* é responsável pelo armazenamento dos dados necessários para o funcionamento da *API de edição*. O esquema lógico que foi usado para a sua implementação é mostrado na Figura 4.

Figura 4 – Esquema do banco de dados



Fonte: Elaborado pelo autor

Como é possível ver na Figura 4, o esquema lógico do *banco de dados de edição* é composto pelas seguintes tabelas:

- **medicament**: representa um medicamento, guardando seu nome, forma e composição ou concentração;

- ***distribution_unit***: representa uma unidade de distribuição de medicamentos, guardando seu nome, localização, horário de funcionamento e uma referência para o seu gerente regional;
- ***medicament_distribution_unit***: representa um medicamento distribuído na unidade, armazenando uma referência a um dos medicamentos cadastrados, uma referência à unidade na qual é distribuído, a quantidade disponível atualmente e se esse medicamento já foi distribuído pelo menos uma vez nessa unidade;
- ***report***: representa uma reclamação feita sobre um medicamento de uma unidade, mantendo o email do criador da reclamação, o conteúdo da reclamação, sua privacidade, a data em que foi feita, uma referência ao medicamento da unidade, a réplica feita por um atendente, e uma referência para o mesmo;
- ***employee***: representa o administrador ou um gerente regional, mantendo nome, login, senha, as áreas sobre as quais ele possui autoridade, seu nível de permissão, e uma referência à seu superior direto, sendo o administrador o único que não possui valor para esse atributo;
- ***attendant***: representa um atendente, guardando seu nome, login, senha e uma referência para a unidade a qual ele está associado;
- ***history***: representa o status de um medicamento em uma unidade em algum momento no passado, mantendo uma referência a um medicamento de uma unidade, a data a qual se refere e a quantidade do medicamento e de suas reclamações feitas nesse dia.

Para prover informações detalhadas sobre o esquema lógico de dados, o dicionário de dados das relações mostradas na Figura 4 está disposto no Apêndice A deste documento.

A lista de medicamentos citada no requisito RF2 é representada pela tabela *medicament* e seria inicialmente povoada com os medicamentos listados na Relação Nacional de Medicamentos Essenciais (Rename), mas a única representação encontrada dessa lista de medicamentos estava em uma cópia do próprio Rename no formato PDF e foram encontradas dificuldades em analisar seu conteúdo programaticamente. Então, como alternativa, foram utilizados os dados da tabela de medicamentos CATMAT¹, que lista os medicamentos presentes no CATMAT, que é o Catálogo de Materiais do Sistema Integrado de Administração de Serviços Gerais (SIASG) do Ministério da Economia (ME).

¹ https://integracao.esusab.ufsc.br/ledi/documentacao/referencias/tabela_catmat.html

3.2 IMPLEMENTAÇÃO

Esta seção descreve a implementação dos componentes que compõem a arquitetura do *Remédio Fácil*. Inicialmente, são apresentados a implementação dos módulos de edição e espacial, e alguns dos endpoints de suas APIs. Em seguida, é mostrado o *módulo de visão*, onde também são explicadas as características de sua implementação, além de algumas das telas que ele apresenta.

3.2.1 Módulo de Edição

Este módulo é encarregado de armazenar e recuperar todas as informações fornecidas pelos usuários. Na sua implementação foi utilizada a linguagem *TypeScript* (BIERMAN; TORGERSEN, 2014), que é uma linguagem de programação baseada na linguagem *JavaScript*, adicionando a ela mais recursos de tipagem de dados. O código escrito em *TypeScript* foi traduzido para a linguagem *JavaScript* e executado com a ferramenta *NodeJs* (TILKOV; VINOSKI, 2010), que é um ambiente de execução que permite executar código escrito na linguagem *JavaScript* sem a necessidade de um navegador.

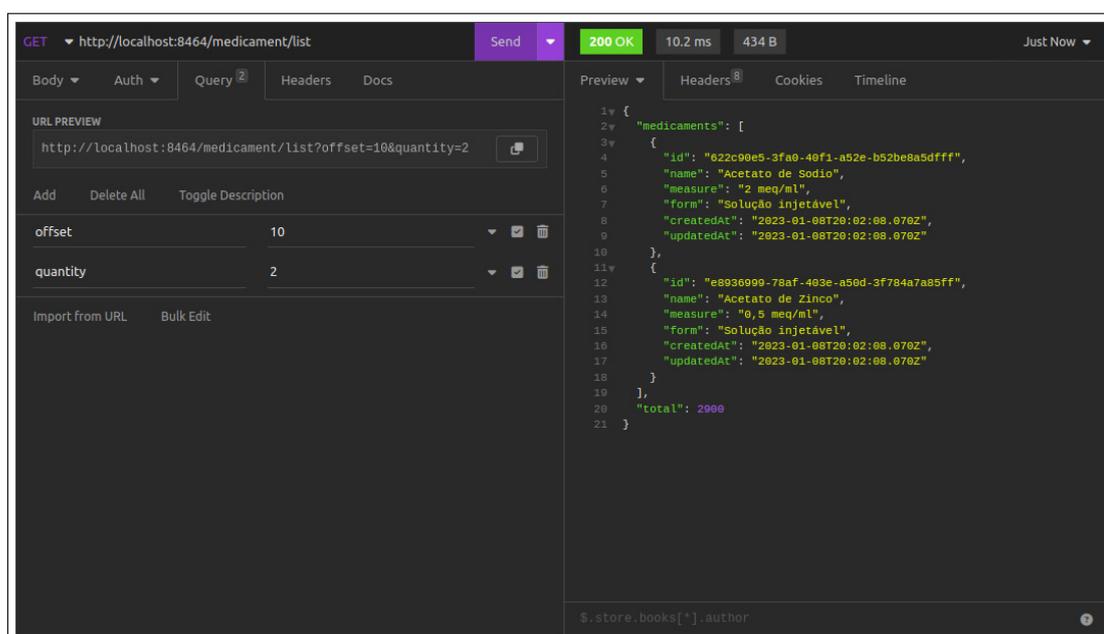
Além disso, o ORM *Sequelize* foi utilizado para estabelecer a comunicação com o banco de dados *PostgreSQL*, enquanto o *framework Express* foi utilizado para abstrair parte dos processos necessários para a recepção e resposta de requisições e para a criação dos endpoints necessários para a comunicação com o *módulo de visão*.

As respostas retornadas pela *API de edição* são baseadas no formato JSON (JavaScript Object Notation) (BRAY, 2017), que também é o único formato aceito no corpo das requisições. Além disso, a API usa *bearer tokens* para realizar a autenticação dos usuários, nas rotas em que essa ação é necessária. A descrição detalhada de todos os *endpoints* da *API de edição* pode ser encontrada no apêndice B deste documento. A seguir, são listados alguns *endpoints* desse módulo e uma breve descrição de cada um deles:

- ***/employee/login***: recebe os dados de autenticação de um usuário e retorna o token necessário para a autenticação em algumas outras rotas;
- ***/employee/registerEmployee***: adiciona um gerente de região;
- ***/medicaments/list***: retorna os medicamentos cadastrados;
- ***/employee/getSubordinates***: retorna os subordinados diretos de um gerente.

A Figura 5 mostra uma requisição feita para o endpoint `/medicament/list` utilizando o *Insomnia*², uma aplicação desktop de código aberto para a geração de requisições para APIs. No lado esquerdo da imagem é mostrado que, além do *endpoint*, também é enviada uma *queryString* com os parâmetros *offset* e *quantity*, que representam respectivamente a quantidade de resultados ignorados no início da consulta e a quantidade de resultados retornados pela consulta. No lado direito da imagem é mostrado o resultado da requisição, que é um texto no formato JSON com as propriedades *medicaments* e *total*, que informam respectivamente os medicamentos solicitados na requisição e o número total de medicamentos armazenados no sistema, que é uma informação útil para questões de paginação.

Figura 5 – Requisição para a rota `/medicament/list`



Fonte: Elaborado pelo autor

3.2.2 Módulo espacial

O *módulo espacial* tem por objetivo o uso de dados espaciais que representam as áreas dos estados e das cidades brasileiras. Esses dados foram extraídos a partir de um conjunto de dados fornecido pelo *OpenStreetMap*. Pelos mesmos motivos citados na descrição do módulo de edição, esse módulo também foi construído utilizando a linguagem *TypeScript*, a ferramenta *NodeJs* e o *framework Express* e também recebe

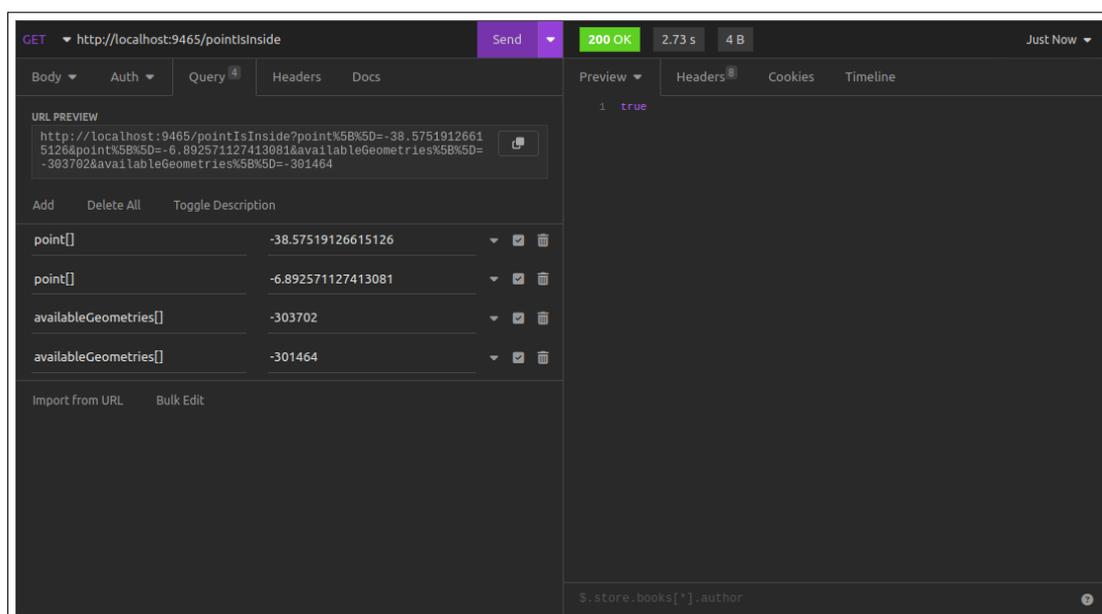
² <https://insomnia.rest>

o corpo das requisições e as responde no formato JSON. Entretanto, diferentemente do *módulo de edição*, ele utiliza a interface de cliente *pg* para se comunicar com o banco de dados *PostgreSQL*. Todos os *endpoints* da *API espacial* são descritos em mais detalhes no apêndice C deste documento. Abaixo são listados alguns *endpoints* desse módulo e uma breve descrição de cada um deles:

- ***/isInside***: verifica se um conjunto de estados e/ou cidades está contido em outro conjunto de estados e/ou cidades;
- ***/pointIsInside***: verifica se um ponto está dentro de um conjunto de estados e/ou cidades;
- ***/getUfs***: retorna o GEOJSON de um ou mais estados;
- ***/getCities***: retorna o GEOJSON de uma ou mais cidades.

A Figura 6 mostra uma requisição feita para o *endpoint* */pointIsInside* utilizando a ferramenta *Insomnia*. No lado esquerdo da imagem é observado que, junto ao *endpoint*, também é enviada uma *queryString* com os parâmetros *point* e *availableGeometries*. O parâmetro *point* é um vetor com dois itens, que correspondem, respectivamente, à latitude e à longitude de um ponto. Já o parâmetro *availableGeometries* é um vetor contendo os identificadores das cidades e/ou estados. No lado direito da imagem é apresentada a resposta da requisição, que é o valor booleano *verdadeiro*, que indica que o ponto representado pelo parâmetro *point* está localizado dentro de pelo uma das geometrias representadas pelo parâmetro *availableGeometries*.

Figura 6 – Requisição para a rota /pointIsInside



Fonte: Elaborado pelo autor

3.2.3 Módulo de visão

O *módulo de visão* tem a responsabilidade de fazer a ligação entre a ferramenta e o usuário final, de modo a tornar essa interação possível mesmo para um usuário sem conhecimento na área de programação.

Para a criação deste módulo foi utilizada a *React*³, que é uma biblioteca baseada na linguagem de programação *JavaScript*. Ela foi escolhida tanto por sua desenvoltura na criação de visualizações interativas e recursos de componentização nativos quanto por ser uma tecnologia já utilizada pelo desenvolvedor da ferramenta.

Em conjunto com a biblioteca *React* foi utilizada a biblioteca *OpenLayers*, descrita na seção 2.2. A biblioteca *OpenLayers* foi escolhida com o objetivo de tornar o uso de dados espaciais interativo e de fácil entendimento para o usuário. Outra tecnologia usada para a implementação deste módulo foi a *Material-UI*⁴, que é uma biblioteca *JavaScript* de componentes para *React*, por disponibilizar componentes prontos e, portanto, acelerar o processo de desenvolvimento. Finalmente, a *Apache ECharts*⁵, que é uma biblioteca *JavaScript* para geração de gráficos, foi usada para

³ <https://react.dev>

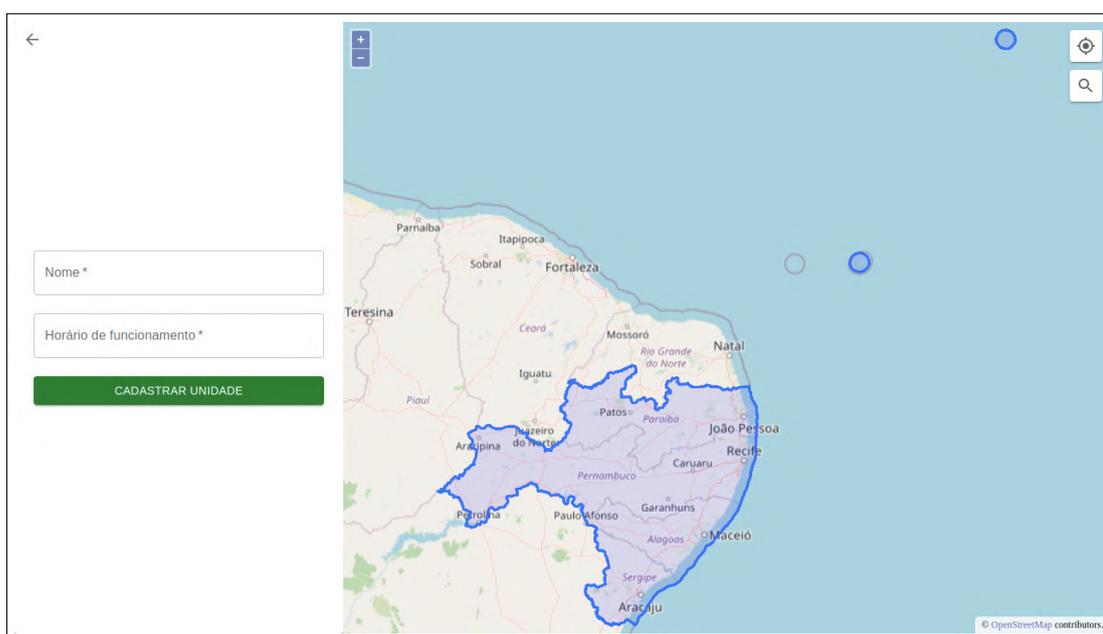
⁴ <https://mui.com>

⁵ <https://echarts.apache.org>

organizar alguns conjuntos de dados de modo a facilitar a sua interpretação por seres humanos. A seguir são mostradas algumas telas presentes neste módulo:

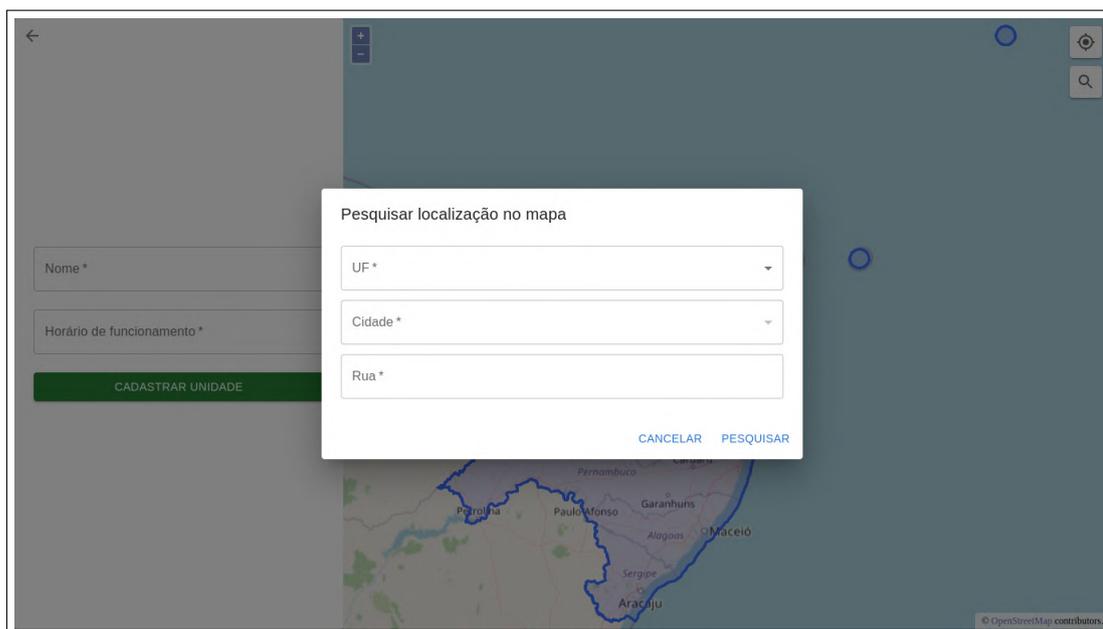
A Figura 7 exibe a tela que permite que um gerente cadastre uma unidade de distribuição de medicamentos em um local dentro de sua área de autoridade. Para isso, ele deve preencher os campos correspondentes ao nome e ao horário de funcionamento da unidade a ser cadastrada, e selecionar um ponto dentro de sua área de autoridade (área coberta em azul escuro no mapa).

Figura 7 – Tela de cadastro de unidade



Fonte: Elaborado pelo autor

Para facilitar o uso do mapa por parte do usuário, também são disponibilizados dois botões no canto superior direito do mapa. O primeiro botão ajusta o foco do mapa para a localização atual do usuário, enquanto que o segundo abre um modal que permite a pesquisa por um endereço no mapa. A Figura 8 mostra o modal que é exibido quando o usuário clica no segundo botão.

Figura 8 – Modal de pesquisa por endereço

Fonte: Elaborado pelo autor

A Figura 9 mostra a tela de criação de um subordinado, que permite que um gerente crie um subordinado direto. Para isso, ele deve informar um nome, email, senha, e escolher um dos níveis de permissão disponíveis. Além disso, também é necessário definir a área de autoridade do novo subordinado. Para tal, é necessário selecionar uma ou mais das áreas cobertas de azul escuro no mapa posicionado do lado direito da tela. Essas áreas correspondem à área de autoridade do gerente e, ao serem selecionadas se tornam opacas com uma cor próxima ao branco. A Figura 9 mostra a tela de criação de subordinado de um gerente com autoridade sobre os estados da Paraíba, Pernambuco, Alagoas e Sergipe, sendo que os dois últimos estão selecionados.

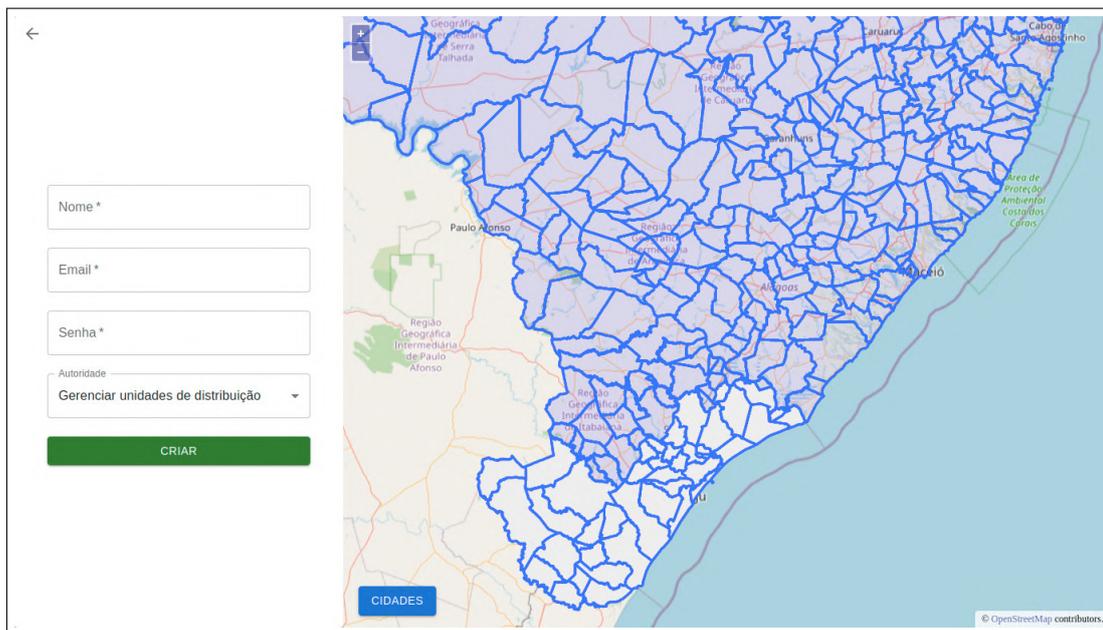
Figura 9 – Tela de criação de subordinado

Fonte: Elaborado pelo autor

No mapa também está disponível uma opção de troca de modo de visualização, que é um botão escrito *Estados* ou *Cidades*, dependendo do modo atualmente ativo. Essa troca de modo visa possibilitar que um gerente crie subordinados com autoridade sobre um conjunto de estados ou um conjunto de cidades.

A Figura 10 mostra a tela de criação de subordinado com o modo de visualização de cidades ativo. O gerente em questão é o mesmo citado na descrição da Figura 9, tendo autoridade sobre os mesmos estados, porém o foco do mapa propositalmente ignora parte de sua área de autoridade para facilitar a visualização das divisões das cidades.

Figura 10 – Modo de visualização de cidades ativo

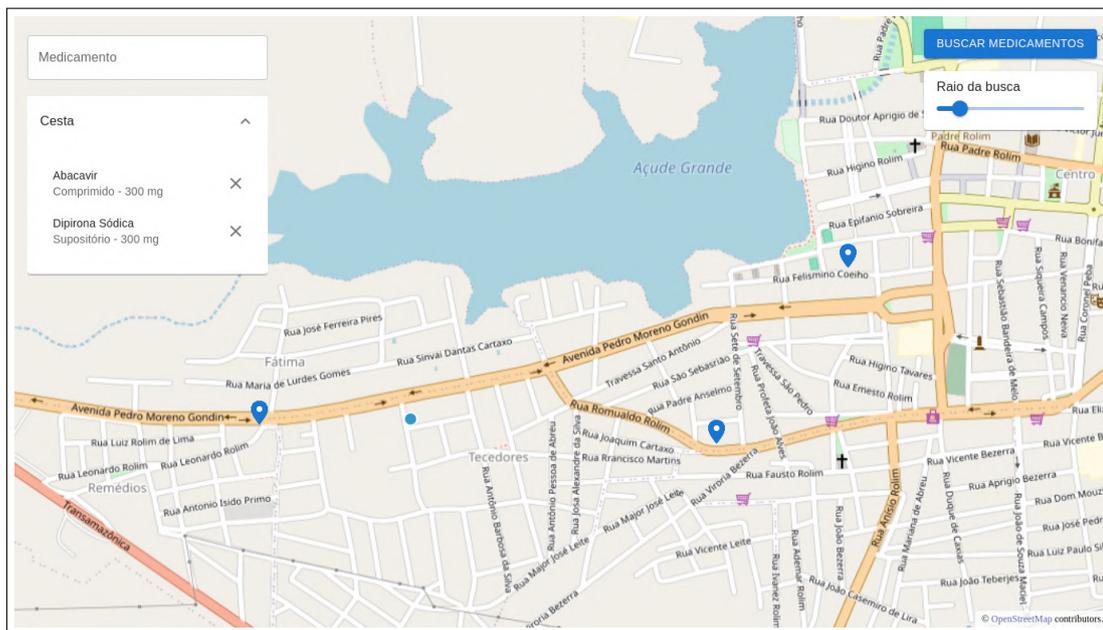


Fonte: Elaborado pelo autor

A Figura 11 mostra a tela de pesquisa de medicamentos, que permite que um usuário não autenticado pesquise pela disponibilidade de medicamentos nas unidades ao redor da sua localização atual. Nela, o campo *Raio da busca* deve ser preenchido com a distância máxima que uma unidade deve ter do ponto atual do usuário para que ela possa ser retornada na busca. Já o campo *Medicamento* permite que o usuário pesquise por medicamentos com base em seus nomes, e selecione os medicamentos resultantes dessa pesquisa. Os medicamentos selecionados são adicionados à *Cesta*, de onde também podem ser removidos ao clicar no ícone de remoção.

Na Figura 11 é possível ver a tela de pesquisa de medicamentos com a *Cesta* preenchida com os medicamentos *abacavir* e *dipirona sódica*.

Figura 11 – Tela de pesquisa de medicamentos

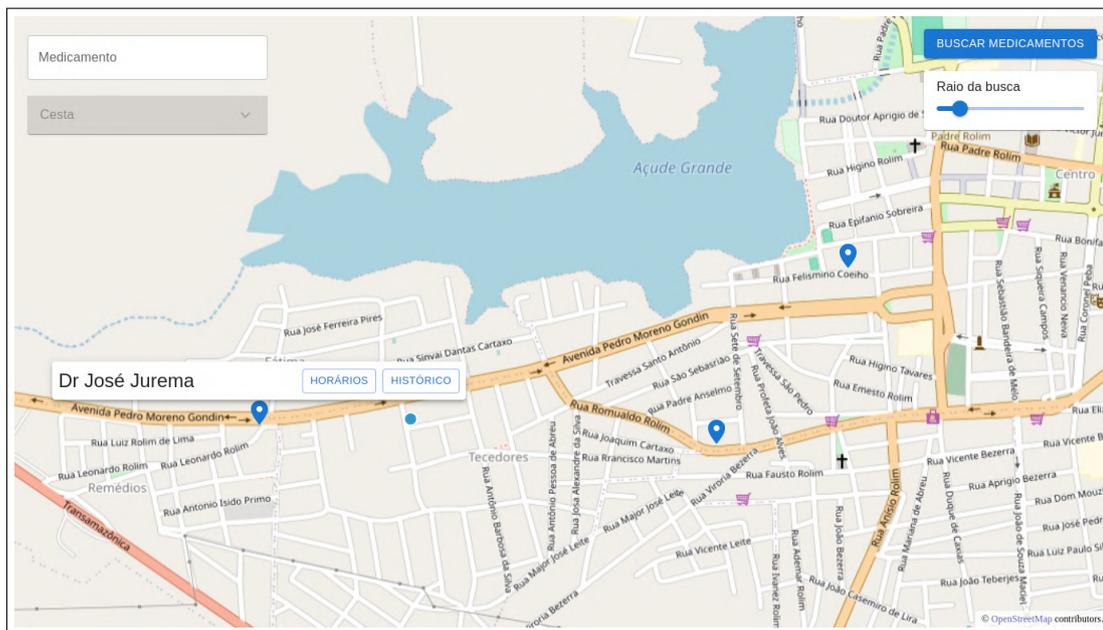


Fonte: Elaborado pelo autor

Ao clicar no botão *Buscar medicamentos*, os medicamentos presentes na *Cesta* são buscados nas unidades dentro do *Raio da busca*, e então o usuário é redirecionado para a tela de resultado de pesquisa de medicamentos.

Como mostrado na Figura 12, nesta tela também é possível clicar nas unidades mostradas no mapa (marcadores azuis). Ao fazer isso, é mostrado o nome da unidade, o botão *Horários*, que abre uma modal com o horário de funcionamento da unidade e o botão *Histórico*, que redireciona para a tela de histórico da unidade.

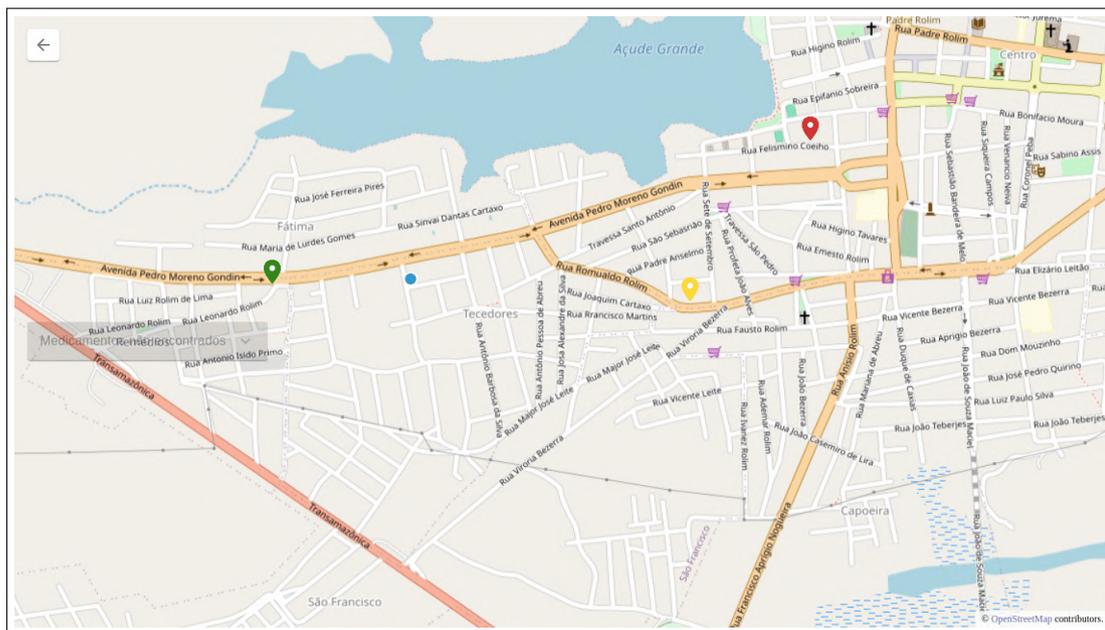
Figura 12 – Informações de uma unidade



Fonte: Elaborado pelo autor

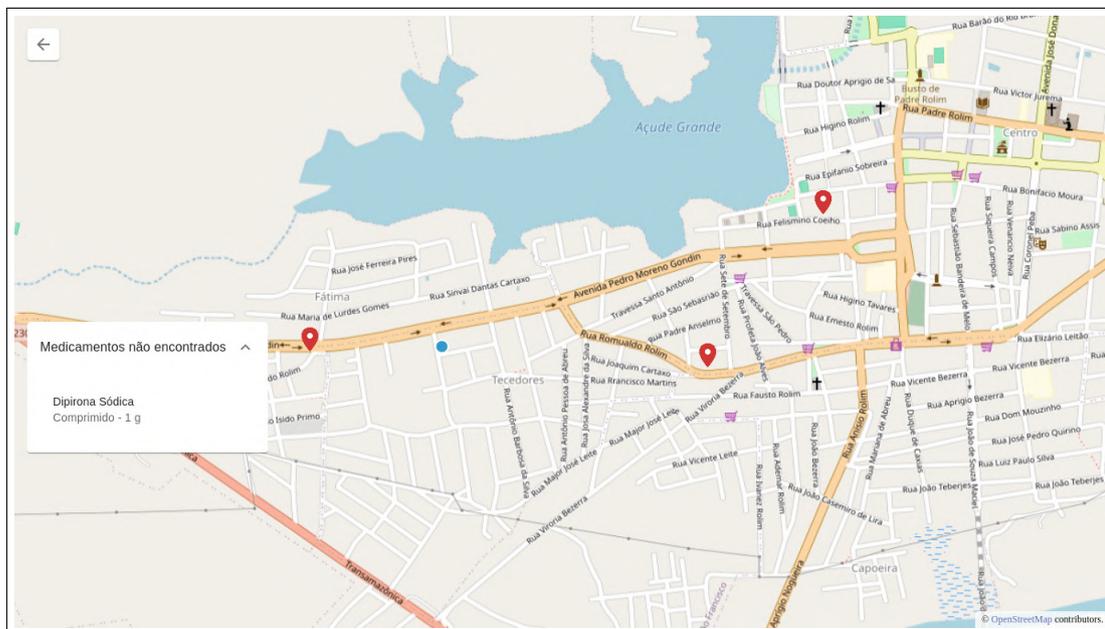
A Figura 13 mostra a tela de resultado de pesquisa de medicamentos, que exibe o resultado da pesquisa feita na tela de pesquisa de medicamentos. Nela, são utilizados marcadores de 3 cores diferentes para representar as unidades dentro do raio da pesquisa: verde, que indica que a unidade possui todos os medicamentos pesquisados; amarelo, que indica que a unidade possui pelo menos um dos medicamentos pesquisados; e vermelho, que indica que a unidade não possui nenhum dos medicamentos pesquisados. Esses três marcadores podem ser vistos nas Figuras 13 e 14.

Figura 13 – Tela de resultado de pesquisa de medicamentos



Fonte: Elaborado pelo autor

Figura 14 – Opção Medicamentos não encontrados

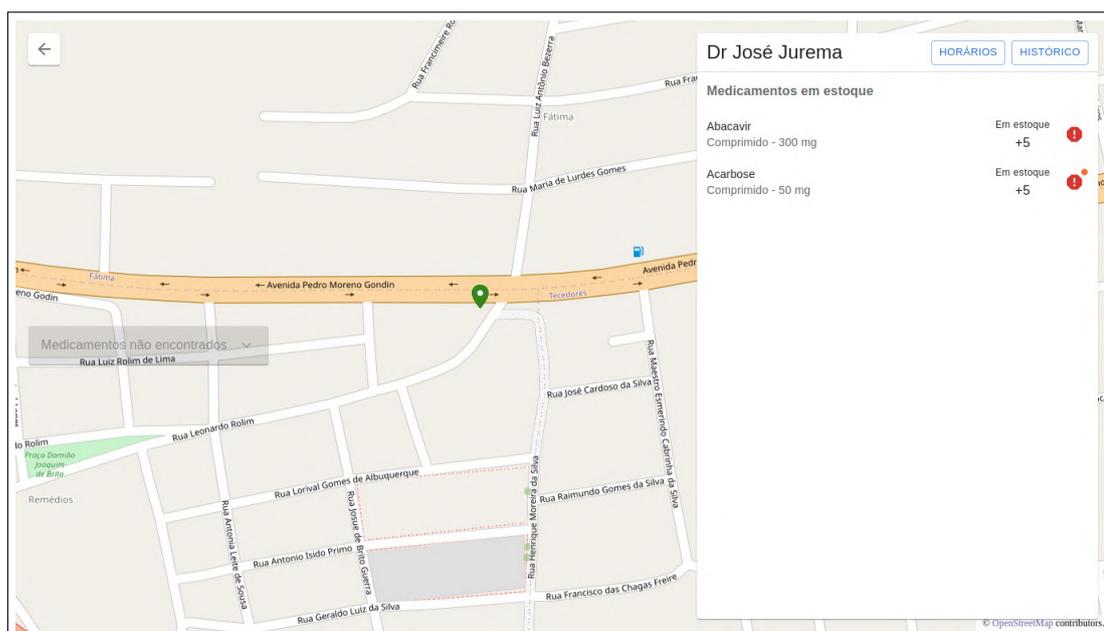


Fonte: Elaborado pelo autor

Nas Figuras 13 e 14 também é possível ver a opção *Medicamentos não encontrados*, que, se disponível, implica que ao menos um dos medicamentos pesquisados não está disponível em nenhuma das unidades dentro do raio da pesquisa. Ao ser aberta, essa opção mostra quais são os medicamentos não encontrados.

Assim como na tela de pesquisa de medicamentos, os marcadores das unidades também podem ser clicados para mostrar suas informações, gerando um resultado como o mostrado na Figura 15. As informações mostradas são as mesmas dessa ação na tela de pesquisa de medicamentos, com adição de até três listas: *Medicamentos em estoque*, onde ficam os medicamentos pesquisados disponíveis nesta unidade; *Medicamentos fora de estoque*, onde ficam os medicamentos pesquisados que estão em falta nesta unidade; e *Medicamentos não distribuídos*, onde ficam os medicamentos pesquisados que nunca foram disponibilizados nesta unidade.

Figura 15 – Informações de uma unidade (Tela de resultado)

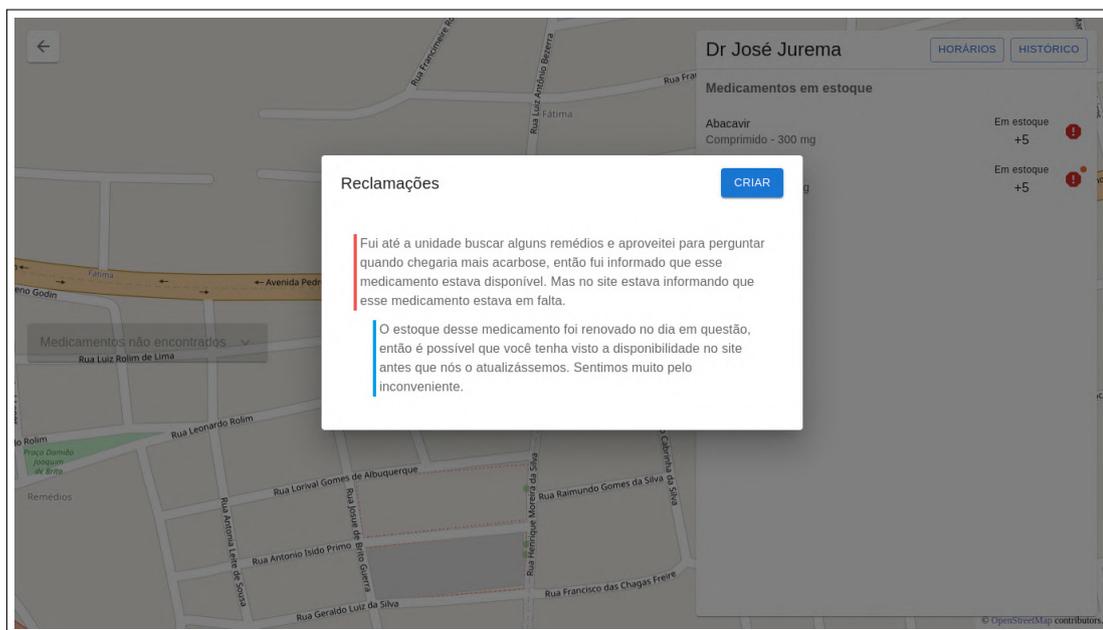


Fonte: Elaborado pelo autor

Cada medicamento nessas listas possui um botão com um símbolo de exclamação, como mostrado na Figura 15. Eles podem ter um círculo laranja em seu canto superior direito, o que indica que o medicamento recebeu pelo menos uma reclamação pública nos últimos trinta dias. Ao acionar este botão, como pode ser visto na Figura 16, é possível ver as reclamações públicas feitas sobre esse medicamento nos últimos

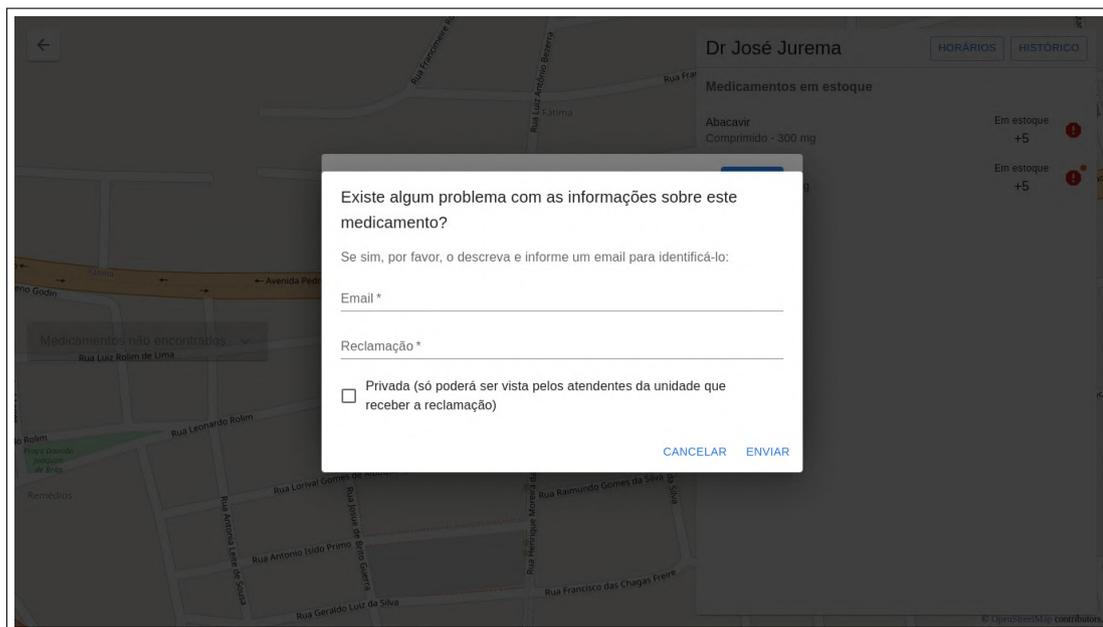
trinta dias e suas respectivas respostas. Também é possível criar novas reclamações sobre o medicamento ao acionar o botão *Criar* e preencher os campos da modal de criação de reclamação, mostrada na Figura 17.

Figura 16 – Reclamações sobre um medicamento



Fonte: Elaborado pelo autor

Figura 17 – Modal de criação de reclamação



Fonte: Elaborado pelo autor

A Figura 18 mostra a tela de histórico do medicamento, que permite que um usuário não autenticado veja o histórico de um medicamento. A visualização do histórico deve ser de um mês específico que pode ser alterado no campo *Mês de referência*. Por padrão, o valor desse campo é o mês que contém o último dia disponível para visualização do histórico, seguindo a restrição do RF17.

Figura 18 – Tela de histórico do medicamento



Fonte: Elaborado pelo autor

O gráfico do canto superior esquerdo mostra a quantidade do medicamento disponível na unidade em cada dia do mês de referência, enquanto o do canto superior direito mostra essa mesma informação organizada em um gráfico de pizza. De modo similar, o gráfico do canto inferior esquerdo mostra a quantidade de reclamações públicas feitas sobre o medicamento em cada dia do mês de referência, enquanto o do canto inferior direito mostra essa mesma informação organizada em um gráfico de pizza.

4 CONSIDERAÇÕES FINAIS

Esta seção compreende as considerações finais deste TCC e é subdividido em conclusão e trabalhos futuros. A seção de conclusão sumariza o resultado do processo descrito neste documento, a ferramenta *Remédio Fácil*, enquanto a seção de trabalhos futuros propõe melhorias para ela.

4.1 CONCLUSÃO

Com o objetivo de facilitar e tornar mais transparente o funcionamento do serviço de distribuição de medicamentos do SUS, este trabalho propôs o *Remédio Fácil*, uma ferramenta que permite o acesso aos dados sobre a distribuição de medicamentos gratuitos a partir de qualquer lugar com internet. Além disso, a ferramenta torna públicos os desacordos e as possíveis inconsistências de informações prestadas pelas unidades, com o uso da mecânica de reclamações.

Como diferencial, a ferramenta ainda guarda as informações de disponibilidade dos medicamentos de cada unidade a partir do dia de sua criação, permitindo que qualquer usuário visualize esses dados, oferecendo, assim, um meio para que a análise da qualidade do serviço de distribuição de medicamentos do SUS possa ser feita utilizando intervalos maiores de tempo.

A ferramenta também proporciona o gerenciamento de unidades de distribuição, atendentes, gerentes e medicamentos, que compõem a base para o funcionamento das mecânicas de visualização de disponibilidade, reclamação e histórico, descritas nos primeiros dois parágrafos desta seção.

4.2 TRABALHOS FUTUROS

Ao final de seu desenvolvimento, a ferramenta *Remédio Fácil* já conta com diversas funcionalidades, mas ainda é possível adicionar novas funcionalidades úteis para o papel que espera-se que esse sistema cumpra.

Por exemplo, uma funcionalidade que permita ao usuário, ao não encontrar os medicamentos de que precisa na unidade de distribuição mais próxima de sua localização atual, informar a locomoção feita na busca de seus medicamentos e se, ao final dessa busca, conseguiu obtê-los em outra unidade de distribuição, em uma farmácia privada, ou se não foi capaz de adquiri-los. Uma funcionalidade como essa

possibilitaria a utilização desse dado para ajudar no processo de tomada de decisão. Por exemplo, ao constatar que muitos habitantes da *cidade A* frequentemente se locomovem até uma unidade de distribuição de medicamentos da cidade vizinha, a *cidade B*, em busca de um determinado medicamento, seria possível embasar medidas como a compra de mais desses medicamentos pela *cidade A*, tanto para evitar a locomoção excessiva dos habitantes que necessitam do medicamento, como para diminuir a pressão sob a unidade da segunda cidade.

Outra funcionalidade que seria útil para o escopo do projeto, mais especificamente para seu objetivo de transparecer o estado da distribuição de medicamentos, é a comparação dos dados armazenados no histórico dos medicamentos com dados externos referentes à compra ou recebimento desses medicamentos. Essa funcionalidade deixaria claro para a população quantos medicamentos uma unidade de distribuição deveria ter recebido e se essa informação está de acordo com as quantidades disponíveis na unidade.

Por último, outra funcionalidade que seria de grande utilidade para os usuários do *Remédio Fácil* seria o agendamento da retirada de medicamentos, similar ao disponível no aplicativo *Remédio Agora*, citado na seção 1.2. Uma funcionalidade desse gênero poderia assegurar aos usuários do sistema que os medicamentos que procuram não saíam de estoque durante seu trajeto até a unidade de distribuição, melhorando, assim, a experiência do usuário.

REFERÊNCIAS

- BENNETT, J. **OpenStreetMap**. [S.l.]: Packt Publishing Ltd., 2010.
- BIERMAN, M. A. G.; TORGERSEN, M. **Understanding TypeScript**. 2014. Disponível em: <<https://users.soe.ucsc.edu/~abadi/Papers/FTS-submitted.pdf>>. Acesso em: 27 de set. 2023.
- BRASIL. [Constituição (1988)]. **Constituição da República Federativa do Brasil**. Brasília, DF: Presidência da República, [2022]. Disponível em: <https://www.planalto.gov.br/ccivil_03/constituicao/constituicao.htm>. Acesso em: 09 de jul. 2023.
- BRAY, T. **The JavaScript Object Notation (JSON) Data Interchange Format**. 2017. Disponível em: <<https://www.rfc-editor.org/rfc/rfc8259>>. Acesso em: 09 de jul. 2023.
- BUTLER, H.; DALY, M.; DOYLE, A.; GILLIES, S.; SCHAUB, T.; HAGEN, S. **The GeoJSON Format**. RFC Editor, 2016. RFC 7946. (Request for Comments, 7946). Disponível em: <<https://www.rfc-editor.org/info/rfc7946>>.
- DATAFOLHA. **Opinião dos brasileiros sobre o atendimento público na área de saúde**. 2018. Disponível em: <https://portal.cfm.org.br/images/PDF/datafolha_sus_cfm2018.pdf>. Acesso em: 19 de jul. 2023.
- FUNDAÇÃO FHC. **Saúde no Brasil: a evolução das políticas e os debates do sistema de saúde nas últimas três décadas**. 2023. Disponível em: <https://fundacaofhc.org.br/linhasdotempo/saude/?gclid=Cj0KCQjwkqSIBhDaARIsAFJANkiPmOI1GDx_Qu8AnYnM0aObdJku83H0A7vDH9j8hFeF808gx8Bg3RgaAudIEALw_wcB>. Acesso em: 09 de jul. 2023.
- GOV.BR. **Sistema Único de Saúde**. s.d. Disponível em: <<https://www.gov.br/saude/pt-br/assuntos/saude-de-a-a-z/s/sus#:~:text=No%20SUS%2C%20a%20responsabilidade%20pela,financeiras%20para%20exercer%20esta%20fun%C3%A7%C3%A3o.>>>. Acesso em: 10 de mar. 2024.
- HAKLAY, M.; WEBER, P. Openstreetmap: User-generated street maps. **IEEE Pervasive Computing**, v. 7, n. 4, p. 12–18, 2008.
- HAZZARD, E. **OpenLayers 2.10 Beginner's Guide**. Birmingham: Packt Publishing Ltd., 2011.
- Secretaria Municipal da Saúde de São Paulo. **Manual de Instruções do Site/Aplicativo Aqui Tem Remédio**. s.d. Disponível em: <[https://www.prefeitura.sp.gov.br/cidade/secretarias/upload/saude/2015_12_16_MANUAL%20AQUI%20TEM%20REMEDIO%20\(1\).pdf](https://www.prefeitura.sp.gov.br/cidade/secretarias/upload/saude/2015_12_16_MANUAL%20AQUI%20TEM%20REMEDIO%20(1).pdf)>. Acesso em: 10 de mar. 2024.
- TILKOV, S.; VINOSKI, S. Node.js: Using javascript to build high-performance network programs. **IEEE Internet Computing**, v. 14, n. 6, p. 80–83, 2010.

Tribunal de Contas do Estado de São Paulo. **1ª Fiscalização Ordenada de 2023 - Estratégia Saúde da Família**. 2023. Disponível em: <https://www.tce.sp.gov.br/sites/default/files/noticias/Relat%C3%B3rio%20Geral%20Consolidado_0.pdf>. Acesso em: 10 de mar. 2024.

WERNECKE, J. **The KML Handbook: Geographic Visualization for the Web**. [S.l.]: Pearson Education, 2008.

APÊNDICE A – DICIONÁRIO DE DADOS

Quadro 1 – Relação medicament

medicament: Relação que armazena os medicamentos cadastrados no sistema			
Atributo	Descrição	Tipo	Restrições
id	Identificador do medicamento	UUID	• Chave primária.
name	Nome do medicamento	TEXT	• Não nulo.
measure	Composição ou concentração do medicamento	TEXT	• Não nulo.
form	Forma do medicamento	TEXT	• Não nulo.

Fonte: Elaborado pelo autor

Quadro 2 – Relação distribution_unit

distribution_unit: Relação que armazena as unidades de distribuição de medicamentos cadastradas no sistema			
Atributo	Descrição	Tipo	Restrições
id	Identificador da unidade	UUID	• Chave primária.
name	Nome da unidade	TEXT	• Não nulo.
location	Localização da unidade	POINT	• Não nulo.
openingHours	Horário de funcionamento da unidade	TEXT	• Não nulo.

Fonte: Elaborado pelo autor

Quadro 3 – Relação `medicament_distribution_unit`

medicament_distribution_unit: Relação que armazena a ligação entre uma unidade de distribuição e um medicamento			
Atributo	Descrição	Tipo	Restrições
<code>id</code>	Identificador da ligação	UUID	<ul style="list-style-type: none"> • Chave primária.
<code>medicamentId</code>	Identificador do medicamento que compõe essa ligação	UUID	<ul style="list-style-type: none"> • Não nulo; • Chave estrangeira para o atributo <i>id</i> da relação <i>medicament</i>.
<code>distributionUnitId</code>	Identificador da unidade que compõe essa ligação	UUID	<ul style="list-style-type: none"> • Não nulo; • Chave estrangeira para o atributo <i>id</i> da relação <i>distribution_unit</i>.
<code>available</code>	Informa se o medicamento já foi disponibilizado pela unidade	BOOLEAN	<ul style="list-style-type: none"> • Não nulo.
<code>quantity</code>	Quantidade atual do medicamento na unidade	INTEGER	

Fonte: Elaborado pelo autor

Quadro 4 – Relação report

report: Relação que armazena as reclamações feitas aos medicamentos nas unidades			
Atributo	Descrição	Tipo	Restrições
id	Identificador da reclamação	UUID	• Chave primária.
medicament Instancelid	Identificador do medicamento que sofreu a reclamação	UUID	• Não nulo; • Chave estrangeira para o atributo <i>id</i> da relação <i>medicament_distribution_unit</i> .
creator	E-mail do usuário que fez a reclamação	TEXT	• Não nulo.
createdAt	Data em que foi feita a reclamação	TIMESTAMP WITH TIME ZONE	• Não nulo.
content	Conteúdo da reclamação	TEXT	• Não nulo.
private	Informa se a reclamação é pública ou privada	BOOLEAN	
replySenderId	Identificador do atendente que respondeu a reclamação	UUID	• Chave estrangeira para o atributo <i>id</i> da relação <i>attendant</i> .
reply	Resposta da reclamação	TEXT	

Fonte: Elaborado pelo autor

Quadro 5 – Relação employee

employee: Relação que armazena as gerentes regionais cadastrados no sistema			
Atributo	Descrição	Tipo	Restrições
id	Identificador do gerente regional	UUID	• Chave primária.
superiorId	Identificador do superior direto	UUID	• Chave estrangeira para o atributo <i>id</i> da relação <i>employee</i> .
name	Nome do gerente regional	TEXT	• Não nulo.
login	Login do gerente regional	TEXT	• Não nulo. • Único.
password	Senha do gerente regional	TEXT	• Não nulo.
authority	Nível de autoridade do gerente regional	SMALLINT	• Não nulo.
managedAreas	Identificadores dos estados e cidades sob autoridade do gerente regional	INTEGER[]	

Fonte: Elaborado pelo autor

Quadro 6 – Relação attendant

attendant: Relação que armazena atendentes cadastrados no sistema			
Atributo	Descrição	Tipo	Restrições
id	Identificador do atendente	UUID	• Chave primária.
distributionUnitId	Identificador da unidade em que o atendente faz parte	UUID	• Não nulo; • Chave estrangeira para o atributo <i>id</i> da relação <i>distribution_unit</i> .
name	Nome do atendente	TEXT	• Não nulo.
login	Login do atendente	TEXT	• Não nulo; • Único.
password	Senha do atendente	TEXT	• Não nulo.

Fonte: Elaborado pelo autor

Quadro 7 – Relação history

history: Relação que armazena o histórico de disponibilidade dos medicamentos de cada unidade			
Atributo	Descrição	Tipo	Restrições
id	Identificador de um registro do histórico	UUID	• Chave primária.
quantity	Quantidade do medicamento disponível na data do registro	INTEGER	• Não nulo.
numberOfReports	Quantidade de reclamações feitas sobre o medicamento na data do registro	INTEGER	• Não nulo.
createdAt	data em que foi feito o registro	TIMESTAMP WITH TIME ZONE	• Não nulo.
instanceId	Identificador do medicamento representado no registro	UUID	• Chave estrangeira para o atributo <i>id</i> da relação <i>medicament_distribution_unit</i> .

Fonte: Elaborado pelo autor

APÊNDICE B – ENDPOINTS DA API DE EDIÇÃO

- POST /employee/login
 - **Descrição:** Recebe os dados de autenticação de um gerente de região e retorna o token necessário para autenticação em algumas outras rotas;
 - **Corpo da requisição:**
 - * **login:** e-mail do usuário;
 - * **password:** senha do usuário.

- POST /employee/registerEmployee
 - **Descrição:** Adiciona um gerente de região;
 - **Autenticação:** Essa rota requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Corpo da requisição:**
 - * **name:** nome do novo gerente;
 - * **login:** email do novo gerente;
 - * **password:** senha do novo gerente;
 - * **authority:** Nível de autoridade do novo gerente. 1 Para permissão de gerenciar unidades de distribuição e 2 para, além disso, ter permissão de gerenciar outros gerentes;
 - * **managedAreas:** Identificadores das áreas sobre as quais o novo gerente tem autoridade.

- PUT /employee/updateSubordinate/:subordinateld
 - **Descrição:** Atualiza o nível de permissão e/ou a área de autoridade de um subordinado direto;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes; região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **subordinateld:** Identificador do subordinado.
 - **Corpo da requisição:**

- * **authority:** Novo nível de permissão do subordinado;
 - * **managedAreas:** Nova área de autoridade do subordinado.

- GET /employee/getManagedAreas
 - **Descrição:** Retorna a área de autoridade do gerente;
 - **Autenticação:** Requer autenticação de um gerente de região.

- GET /employee/distributionUnit
 - **Descrição:** Retorna as unidades de distribuição do gerente;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de consulta:**
 - * **offset:** número de unidades que devem ser ignoradas;
 - * **limit:** número máximo de unidades que devem ser retornadas.

- GET /employee/getSubordinates
 - **Descrição:** Retorna os subordinados diretos do gerente;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de consulta:**
 - * **offset:** número de subordinados que devem ser ignorados;
 - * **limit:** número máximo de subordinados que devem ser retornados.

- GET /employee/getSubordinates/:guid
 - **Descrição:** Retorna os dados de um subordinado direto com base em seu identificador;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **guid:** Identificador do subordinado.

- GET /employee/getInfo

- **Descrição:** Retorna os dados do gerente;
 - **Autenticação:** Requer autenticação de um gerente de região.

- PUT /employee
 - **Descrição:** Atualiza o nome do gerente;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Corpo da requisição:**
 - * **name:** Novo nome do gerente.

- PUT /employee/updatePassword
 - **Descrição:** Atualiza a senha do gerente;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Corpo da requisição:**
 - * **oldPassword:** Senha atual do gerente;
 - * **newPassword:** Nova senha do gerente.

- DEL /employee/removeSubordinate/:subordinateld
 - **Descrição:** remove um subordinado direto;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **subordinateld:** Identificador do subordinado.

- POST /employee/registerAttendant/:distributionUnitld
 - **Descrição:** Adiciona um atendente a uma unidade de distribuição sobre a qual tenha posse;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de caminho:**
 - * **distributionUnitld:** identificador da unidade de distribuição a qual o atendente será adicionado.
 - **Corpo da requisição:**

- * **name:** Nome do atendente;
 - * **login:** Email do atendente;
 - * **password:** Senha do atendente.

- GET /employee/getAttendants/:distributionUnitId
 - **Descrição:** Retorna os atendentes de uma unidade de distribuição sobre a qual tenha posse;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de caminho:**
 - * **distributionUnitId:** Identificador da unidade de distribuição;
 - **Parâmetros de consulta:**
 - * **offset:** número de atendentes que devem ser ignorados;
 - * **limit:** número máximo de atendentes que devem ser retornados.

- DEL /employee/removeAttendant/:attendantId
 - **Descrição:** Remover um atendente de uma unidade de distribuição sobre a qual tenha posse;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de caminho:**
 - * **attendantId:** Identificador do atendente.

- GET /employee/employeesWithinTheManagedAreas/:employeeId
 - **Descrição:** Retorna quais subordinados diretos podem ser transferidos para um subordinado direto em específico;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **employeeId:** Identificador do subordinado para o qual a rota verifica se os outros podem ser transferidos.

- GET /employee/getSubordinateSubordinates/:employeeId
 - **Descrição:** Retorna os subordinados diretos de um subordinado direto;

- **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
- **Parâmetros de caminho:**
 - * **employeeId:** Identificador do subordinado cujos subordinados diretos são retornados.

- POST /employee/exchangeSubordinates/:employeeId
 - **Descrição:** Troca subordinados diretos entre um gerente e um de seus subordinados diretos;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **employeeId:** Identificador do subordinado alvo da troca.
 - **Corpo da requisição:**
 - * **received:** identificadores dos subordinados que serão recebidos pelo gerente;
 - * **transmitted:** identificadores dos subordinados que serão transmitidos para o subordinado alvo.

- POST /attendant/login
 - **Descrição:** Recebe os dados de autenticação de um atendente e retorna o token necessário para autenticação em algumas outras rotas;
 - **Corpo da requisição:**
 - * **login:** e-mail do usuário;
 - * **password:** senha do usuário.

- GET /attendant/getInfo
 - **Descrição:** Retorna os dados do atendente;
 - **Autenticação:** Requer autenticação de um atendente.

- PUT /attendant
 - **Descrição:** Atualiza o nome do atendente;

- **Autenticação:** Requer autenticação de um atendente;
- **Corpo da requisição:**
 - * **name:** Novo nome do atendente.

- PUT /attendant/updatePassword
 - **Descrição:** Atualiza a senha do atendente;
 - **Autenticação:** Requer autenticação de um atendente;
 - **Corpo da requisição:**
 - * **oldPassword:** Senha atual do atendente;
 - * **newPassword:** Nova senha do atendente.

- POST /medicament
 - **Descrição:** Cria um medicamento;
 - **Autenticação:** Requer autenticação de administrador;
 - **Corpo da requisição:**
 - * **name:** Nome do medicamento;
 - * **measure:** composição ou concentração do medicamento;
 - * **form:** forma do medicamento.

- GET /medicament/search
 - **Descrição:** Pesquisa medicamentos por nome;
 - **Parâmetros de consulta:**
 - * **search:** Nome ou trecho do nome de um medicamento com no mínimo três caracteres.

- GET /medicament/list
 - **Descrição:** Retorna os medicamentos;
 - **Parâmetros de consulta:**
 - * **offset:** número de medicamentos que devem ser ignorados;
 - * **quantity:** número máximo de medicamentos que devem ser retornados;
 - * **search:** Nome ou trecho do nome de um medicamento que deve estar contido no nome de todos os medicamentos retornados.

- GET /medicament/list/:unitId
 - **Descrição:** Retorna os medicamentos e, para aqueles disponibilizados pela unidade informada, retorna uma referência para o identificador do medicamento dentro dessa unidade;
 - **Parâmetros de caminho:**
 - * **unitId:** identificador da unidade de distribuição.
 - **Parâmetros de consulta:**
 - * **onlyAvailable:** Se verdadeiro, retorna apenas medicamentos disponibilizados pela unidade;
 - * **offset:** número de medicamentos que devem ser ignorados;
 - * **quantity:** número máximo de medicamentos que devem ser retornados;
 - * **search:** Nome ou trecho do nome de um medicamento que deve estar contido no nome de todos os medicamentos retornados.

- GET /medicament/get-medicaments-list
 - **Descrição:** Retorna uma lista de medicamentos correspondente à lista de identificadores enviada;
 - **Autenticação:** Requer autenticação de um atendente;
 - **Parâmetros de consulta:**
 - * **medicaments:** Identificadores de medicamentos da unidade de distribuição do atendente autenticado.

- PUT /medicament/update-stock
 - **Descrição:** Atualiza a quantidade de um ou mais medicamentos da unidade do agente autenticado;
 - **Autenticação:** Requer autenticação de um atendente;
 - **Corpo da requisição:**
 - * **changes:** Array de objetos com o id do medicamento (id) e a alteração em sua quantidade (quantity).

- GET /medicament/search-medicaments
 - **Descrição:** Pesquisa por medicamentos nas unidades de distribuição dentro da área de um círculo;

- **Parâmetros de consulta:**
 - * **medicaments:** Lista de identificadores de medicamentos;
 - * **center:** Ponto inicial da pesquisa;
 - * **radio:** Raio da pesquisa.

- GET /distributionUnit/getAll
 - **Descrição:** Retorna todas as unidades de distribuição.

- GET /distributionUnit/search-units
 - **Descrição:** Retorna as unidades de distribuição dentro da área de um círculo;
 - **Parâmetros de consulta:**
 - * **center:** Ponto inicial da pesquisa;
 - * **radio:** Raio da pesquisa.

- GET /distributionUnit/get-info/:unitId
 - **Descrição:** Retorna as informações de uma unidade;
 - **Parâmetros de caminho:**
 - * **unitId:** Identificador da unidade.

- POST /distributionUnit
 - **Descrição:** Cria uma unidade de distribuição;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Corpo da requisição:**
 - * **name:** Nome da unidade;
 - * **location:** Localização da unidade;
 - * **openingHours:** Horário de funcionamento da unidade.

- GET /distributionUnit/unitsWithinManagedArea/:employeeId
 - **Descrição:** Retorna as unidades do gerente que podem ser transferidas para o subordinado direto especificado;

- **Autenticação:** Essa rota requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
- **Parâmetros de caminho:**
 - * **employeeld:** Identificador do subordinado direto.

- GET /distributionUnit/getSubordinateUnits/:employeeld
 - **Descrição:** Retorna as unidades de um subordinado direto;
 - **Autenticação:** Essa rota requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **employeeld:** Identificador do subordinado direto.

- POST /distributionUnit/exchangeUnits/:employeeld
 - **Descrição:** Troca unidades de distribuição entre um gerente e um de seus subordinados diretos;
 - **Autenticação:** Requer autenticação de um gerente de região com permissão de gerenciar outros gerentes;
 - **Parâmetros de caminho:**
 - * **employeeld:** Identificador do subordinado alvo da troca.
 - **Corpo da requisição:**
 - * **received:** identificadores dos subordinados que serão recebidos pelo gerente;
 - * **transmitted:** identificadores dos subordinados que serão transmitidos para o subordinado alvo.

- GET /distributionUnit/get-medicaments
 - **Descrição:** Retorna uma lista de medicamentos e a quantidade de cada um na unidade do atendente;
 - **Autenticação:** Requer autenticação de um atendente;
 - **Parâmetros de consulta:**
 - * **offset:** número de medicamentos que devem ser ignorados;
 - * **quantity:** número máximo de medicamentos que devem ser retornados;

- * **search:** Nome ou trecho do nome de um medicamento que deve estar contido no nome de todos os medicamentos retornados.

- GET /distributionUnit/:unitGuid
 - **Descrição:** Pegar informações de uma unidade de distribuição;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de caminho:**
 - * **unitGuid:** Identificador da unidade.

- PATCH /distributionUnit/:unitGuid
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de caminho:**
 - * **unitGuid:** Identificador da unidade.
 - **Corpo da requisição:**
 - * **name:** Nome da unidade;
 - * **location:** Localização da unidade;
 - * **openingHours:** Horário de funcionamento da unidade.

- DEL /distributionUnit/:unitGuid
 - **Descrição:** Remove uma unidade de distribuição;
 - **Autenticação:** Requer autenticação de um gerente de região;
 - **Parâmetros de caminho:**
 - * **unitGuid:** Identificador da unidade.

- POST /report
 - **Descrição:** Cria uma reclamação;
 - **Corpo da requisição:**
 - * **creator:** E-mail do criador;
 - * **content:** Conteúdo da reclamação;
 - * **private:** Se a reclamação é privada ou não;
 - * **medicamentId:** Identificador do medicamento atrelado à reclamação;

- * **unitId**: Identificador da unidade de distribuição atrelada à reclamação.

- GET /report
 - **Descrição**: Retorna as reclamações feitas sobre um medicamento de uma unidade nos últimos 30 dias;
 - **Parâmetros de consulta**:
 - * **medicamentId**: Identificador do medicamento;
 - * **unitId**: Identificador da unidade.

- GET /report/last-month
 - **Descrição**: Retorna as reclamações dos últimos trinta dias da unidade de um atendente;
 - **Autenticação**: Requer autenticação de um atendente.

- POST /report/:reportGuid/reply
 - **Descrição**: Responde uma reclamação feita sobre um medicamento da unidade do atendente;
 - **Autenticação**: Requer autenticação de um atendente;
 - **Parâmetros de caminho**:
 - * **reportGuid**: Identificador da reclamação.
 - **Corpo da requisição**:
 - * **content**: Resposta da reclamação.

- GET /history/:instanceld
 - **Descrição**: Retorna o histórico de um medicamento em um mês específico;
 - **Parâmetros de caminho**:
 - * **instanceld**: Identificador do medicamento.
 - **Parâmetros de consulta**:
 - * **month**: Mês do histórico;
 - * **year**: Ano do histórico.

APÊNDICE C – ENDPOINTS DA API ESPACIAL

- GET /pointIsInside
 - **Descrição:** Verifica se um ponto está dentro de um conjunto de estados e/ou cidades;
 - **Parâmetros de consulta:**
 - * **point:** Array de tamanho dois com a latitude e longitude de um ponto;
 - * **availableGeometries:** Array de identificadores de cidades e/ou estados.

- GET /isInside
 - **Descrição:** Verifica se um conjunto de estados e/ou cidades está contido em outro conjunto de estados e/ou cidades;
 - **Parâmetros de consulta:**
 - * **requestedGeometries:** Array com os identificadores das cidades e/ou estados que a rota deve averiguar se estão contidos no outro conjunto;
 - * **availableGeometries:** Array com os identificadores das cidades e/ou estados que a rota deve verificar se contém o outro conjunto.

- POST /getUfs
 - **Descrição:** Retorna o geojson de um ou mais estados;
 - **Corpo da requisição:**
 - * **ids:** identificadores dos estados cujos geojsons devem ser retornados.

- POST /getCities
 - **Descrição:** Retorna o geojson de uma ou mais cidades;
 - **Corpo da requisição:**
 - * **ids:** identificadores das cidades cujos geojsons devem ser retornados.

- POST /oneGeometry
 - **Descrição:** Junta uma ou mais cidades e/ou estados em uma única geometria e retorna o geojson dessa nova geometria;

– **Corpo da requisição:**

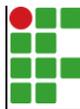
* **ids:** identificadores das cidades e/ou estados.

• GET /summarizeInCities

– **Descrição:** Recebe identificadores de cidades e/ou estados e retorna os identificadores das cidades informadas e os identificadores das cidades dentro dos estados informados;

– **Parâmetros de consulta:**

* **ids:** identificadores das cidades e/ou estados.

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Cajazeiras - Código INEP: 25008978
	Rua José Antônio da Silva, 300, Jardim Oásis, CEP 58.900-000, Cajazeiras (PB)
	CNPJ: 10.783.898/0005-07 - Telefone: (83) 3532-4100

Documento Digitalizado Ostensivo (Público)

Trabalho de Conclusão de Curso

Assunto:	Trabalho de Conclusão de Curso
Assinado por:	Ruan Cardoso
Tipo do Documento:	Anexo
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- **Ruan Miguel Coelho Cardoso, ALUNO (201812010013) DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS - CAJAZEIRAS**, em 14/03/2024 13:15:06.

Este documento foi armazenado no SUAP em 14/03/2024. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1115830
Código de Autenticação: 03abfd50bf

