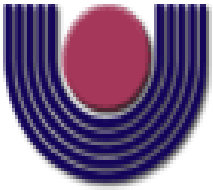


UNIOESTE – Universidade Estadual do Oeste do Paraná
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
Colegiado de Ciência da Computação
Curso de Bacharelado em Ciência da Computação

Apresentação de Conteúdo Geoespacial no Sistema de Informações Geográficas Aedes (SIGAEDES)

Murillo Douglas Oliveira Machado

CASCAVEL
2017

MURILLO DOUGLAS OLIVEIRA MACHADO

APRESENTAÇÃO DE CONTEÚDO GEOESPACIAL NO SISTEMA DE INFORMAÇÕES GEOGRÁFICAS AEDES (SIGAEDES)

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação, do Centro de Ciências Exatas e Tecnológicas da Universidade Estadual do Oeste do Paraná - Campus de Cascavel.

Orientador: Profª. Claudia Brandelero Rizzi.

CASCAVEL

2017

MURILLO DOUGLAS OLIVEIRA MACHADO

APRESENTAÇÃO DE CONTEÚDO GEOESPACIAL NO SISTEMA DE INFORMAÇÕES GEOGRÁFICAS AEDES (SIGAEDES)

Monografia apresentada como requisito parcial para obtenção do Título de Bacharel em Ciência da Computação, pela Universidade Estadual do Oeste do Paraná, Campus de Cascavel. Aprovada pela Comissão formada pelos professores:

_____

Profª. Claudia Brandelero Rizzi (Orientadora)
Colegiado de Ciência da Computação,
UNIOESTE

_____

Prof. Claudio Leones Bazzi (Co-Orientador)
UTFPR Campus Medianeira

_____

Prof. Guilherme Galante
Colegiado de Ciência da Computação
UNIOESTE

_____

Prof. Rogério Luis Rizzi
Colegiado de Ciência da Computação
UNIOESTE

Cascavel, 18 de dezembro de 2017.

"What I am saying then is just because you don't know how you manage to be conscious, how you manage to grow and shape your body, doesn't mean that you're not doing it. Equally, if you don't know how the universe shines the stars, constellates the constellations, or galactifies the galaxies – you don't know but that doesn't mean that you aren't doing it just the same way as you are breathing without knowing how you breathe."

Alan Watts

AGRADECIMENTOS

A Unioeste e seu corpo docente, que me direcionaram durante todos estes anos em que fui aluno da instituição, especialmente a minha orientadora Profª. Claudia Brandelero Rizzi, por ser compreensiva e me aconselhar em vários momentos de dificuldades durante este período.

A minha família, especialmente minha mãe Maria Rosa de Oliveira e minha avó Anezia Aragão Oliveira que sempre estiveram presentes em minha vida me ensinando a ser uma pessoa melhor e me encorajando a não desistir.

Aos incontáveis amigos e colegas que fiz durante esses anos nos trabalhos do curso, projetos de extensão, na Leão (melhor atlética do mundo), na maratona de programação, nos balangas (R.I.P. pessegueiro) e tantas outras atividades que participei e que foram de fundamental importância para a construção da pessoa que sou hoje.

A minha namorada Larissa Drechsler que, apesar dos mais de 600km de distância, consegue se fazer presente diariamente em minha vida, me motivando e ajudando sempre que preciso.

A todos que de alguma forma fizeram parte da minha formação profissional ou pessoal, o meu muito obrigado.

Lista de Figuras

Lista de Tabelas

SUMÁRIO

Resumo

No contexto de combate à Dengue, Zika e Chikungunya o método mais eficiente atualmente ainda é a prevenção e combate ao vetor transmissor, o mosquito *Aedes*. Em Cascavel, município do Paraná, o órgão responsável por ações de controle e combate é o Setor de Controle de endemias. Este setor segue um conjunto de atividades operacionais preconizadas pelo Ministério da Saúde e pela Organização Mundial da Saúde que visam reduzir os danos causados pelas doenças em questão. O objetivo deste trabalho então é apresentar o desenvolvimento de uma ferramenta de gestão que utiliza de consultas e visualizações de dados geoespaciais construídas com tecnologias livres para contribuir como embasamento para a tomada de decisões dos gestores durante o planejamento e execução das atividades operacionais. Esta ferramenta de gestão é o Sistema de Informações Geográficas AEDES (SIGAEDES), e foi construída utilizando a metodologia de prototipação incremental. Suas funcionalidades que necessitam de apresentação de conteúdo geoespacial foram implementadas utilizando Java como linguagem padrão no *server-side*, e JavaScript em conjunto com a biblioteca OpenLayers no *client-side*. Para o banco de dados foi utilizado PostgreSQL em conjunto com sua extenção para bancos de dados geográficos PostGis. Como base para as consultas espaciais foram utilizadas informações disponibilizadas pela Prefeitura Municipal de Cascavel. Como resultado do trabalho, as funcionalidades elicitadas inicialmente juntamente com os funcionários do Setor de Controle de Endemias foram ajustadas, implementadas e verificadas. A implantação do sistema para uso dos agentes não foi concluída a tempo para a finalização deste trabalho, mas será executada como trabalho posterior, visto que durante a etapa de verificação das funcionalidades, os gestores demonstraram grande interesse na utilização do sistema.

Palavras chave: SIGAEDES, sistema de informações, dengue, zika, chikungunya, georeferenciamento.

Capítulo 1

Introdução

A dengue é uma doença infecciosa causada por um arbovírus do gênero Flavivírus da família Flaviviridae e possui quatro diferentes sorotipos de vírus. Cada um dos quatro sorotipos DEN-1, DEN-2, DEN-3 e DEN-4 podem causar enfermidade grave e mortal e a infecção por um deles confere proteção permanente para o mesmo sorotipo e imunidade parcial e temporária contra os outros três, mas a infecção subsequente com um segundo sorotipo aumenta a probabilidade de doença grave como a hemorrágica [1].

Diversas espécies de mosquitos Aedes podem servir como vetores transmissores do vírus da dengue. No Brasil, duas delas estão instaladas, o *Aedes albopictus* e especialmente, o *Aedes aegypti*, um mosquito urbano, de hábito diurno e altamente ativo. As mesmas espécies são vetores de outras doenças como Zika e Chikungunya. A transmissão ocorre quando a fêmea da espécie se contamina ao picar um indivíduo infectado na fase de viremia, tornando-se capaz, após período de 10 a 14 dias, de transmitir o vírus a todos os indivíduos que picar [2], [3].

Os principais sintomas da Dengue são febre alta com inicio súbito, forte dor de cabeça, perda do paladar e apetite, manchas e erupções na pele semelhantes ao sarampo, forte dor de cabeça, náuseas e vômito.

Quanto à Chikungunya, os sintomas podem se caracterizar principalmente por dor no corpo intensa. A recuperação é o resultado esperado, mas a convalescência pode ser prolongada, algumas vezes até um ano ou mais [4]. Embora ainda em confirmação, estudos apontam a correlação entre a infecção por Zika e a ocorrência de síndrome de Guillain-Barré e microcefalia em bebês cujas mães gestantes estavam infectadas

Os principais sintomas de Zika são febre, erupções na pele, dor no corpo e conjuntivite. Sua evolução geralmente é branda e os sintomas duram de três a sete dias [6]. Embora ainda em confirmação, estudos apontam a correlação entre a infecção por Zika e a ocorrência de síndrome de Guillain-Barré e microcefalia em bebês cujas mães gestantes estavam infectadas [7].

Em 2016 foram registrados no Brasil 1.500.535 casos prováveis de dengue, 271.824 casos da febre Chikungunya e 215.319 casos prováveis de febre pelo vírus Zika, e, até metade do mês de abril de 2017 já foram registrados 113.381 casos de dengue, 43.010 casos de Chikungunya e 7.911 casos prováveis de febre pelo vírus Zika [5].

Em decorrência dos problemas causados pelas doenças Dengue, Chikungunya e Zika e agravado pela possibilidade de co-circulação de vírus, co-infecção e infecção sequencial por essas doenças e a falta de um meio efetivo de combate, senão por meio da prevenção, faz-se necessário a disponibilização de mecanismos de otimização de gestão para elevar o nível de acurácia e celeridade na tomada de decisões de gestores em saúde publica em ações de controle e combate vetorial.

Nesse contexto está o desenvolvimento do Sistema de Informações Geográficas Aedes (SIGAEDES). Trata-se de um Sistema de Informações que visa integrar várias metodologias preconizadas pelo Ministério da Saúde e pela Organização Mundial de Saúde, além daquelas operacionais realizadas pelo setor específico do município. Dentre as rotinas que o SIGAEDES viabiliza estão àquelas referentes ao geoprocessamento de dados inerentes às ações práticas que dá suporte. Essas rotinas envolvem a visualização de fatos como a localização de pontos estratégicos, indivíduos com suspeita ou confirmação da doença e ações realizadas pelo pessoal responsável pelo controle e combate ao vetor que, em Cascavel é realizado pelo Controle de Endemias, vinculado à Secretaria Municipal de Saúde.

Atualmente o trabalho de combate e prevenção dos vetores de Dengue, Zika e Chikungunya realizado pelo Controle de Endemias de Cascavel possui como suporte apenas formulários em papel e croquis desenhados e atualizados a mão. Esses croquis possuem informações dos lotes de toda a cidade de Cascavel e servem como base para o planejamento de algumas ações operacionais de campo das equipes.

E neste sentido, segundo o censo demográfico de 2010 realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o município de Cascavel possui mais de 91 mil domicílios particulares ocupados [8]. Considerando, que os croquis gerados pelos agendes do Controle de Endemias englobam todos os domicílios, sejam ocupados ou não, construções comerciais, terrenos baldios, entre outras categorias de terrenos, manter todas essas informações atualizadas através de trabalho manual se torna custoso, demorado e, geralmente, impreciso.

Portanto as funcionalidades de consultas e visualizações de mapas dentro do SIGAEDES se tornam uma parte essencial do sistema, se centrando na facilidade de visualização de resultados. Além disso, viabilizam sua associação com as áreas geográficas de ocorrência, oferecendo uma nova dimensão das relações que estabelecem, melhorando a qualidade das informações armazenadas, das decisões tomadas pelos supervisores e minimizando o desperdício de tempo de trabalho manual dos agentes, para que possa ser empregado em ações mais importantes.

1.1 Objetivos

O objetivo geral deste trabalho de conclusão de curso é desenvolver soluções de consultas e visualizações no Sistema de Informações Geográficas Aedes (SIGAEDES) que sejam funcionais, eficientes e úteis para o Controle de Endemias, e utilizem tecnologias livres.

Os objetivos específicos que serão realizados para alcançar o objetivo principal deste trabalho contemplam:

a) realizar revisão bibliográfica sobre sistemas de informações geográficos que empreguem tecnologias web, mas necessariamente, OpenLayers, PostgreSQL, PostGIS;

b) identificar e especificar as funcionalidades que requeiram geovisualização no SIGAEDES;

c) implementar e testar as funcionalidades que requeiram geovisualização no SIGAEDES;

d) avaliar e analisar os resultados.

1.2 Metodologia

A metodologia adotada para o desenvolvimento deste projeto se dará por meio de prototipação. Estes protótipos serão construídos por meio de um conjunto de melhorias incrementais, que visam atender os cenários em que os requisitos e tecnologia não são totalmente conhecidos, são voláteis e os *stakeholders* necessitam de versões parciais do futuro sistema. O Sistema foi projetado para gerar produtos de qualidade, que atendam os requisitos elicitados, e que possibilitem sua manutenção e extensibilidade [10], [11], [12].

As tecnologias utilizadas serão o banco de dados PostGreSQL [13] com a extensão espacial PostGIS [14] em conjunto com as linguagens de programação Java [56], JavaScript [16] e as bibliotecas JQuery [17] e OpenLayers[18]; a linguagem de marcação HTML5 [19], além do CSS [20]; O formato para troca de dados é o JSON [21] e são também utilizados o Play Framework [22] e o Bootstrap [23]. Os aplicativos para dispositivos móveis Android serão desenvolvidos com Java [15], XML [24] e SqLite [25].

A avaliação dos resultados das consultas e soluções implementadas se dará por meio de um estudo de caso a partir da utilização do sistema pelos agentes e supervisores do Controle de Endemias de Cascavel. Serão elaborados e aplicados questionários para coleta de informações sobre desempenho e qualidade das soluções de geovisualizações em representantes específicos dos *stakeholders.*

Capítulo 2

Geoinformação, Sistemas de Informação Geográfica e Tecnologias WEB para Geoprocessamento

A Ciência da Geoinformação tem como principal objetivo utilizar um conjunto de técnicas computacionais e matemáticas para coletar e tratar informações espaciais para um objetivo específico. Inúmeras áreas do conhecimento utilizam informações espaço-temporais para estudo de fenômenos ambientais e urbanos. Segundo Câmara [27], “Quando falamos que o espaço é uma linguagem comum no uso de SIG, estamos nos referindo ao espaço computacionalmente representado e não aos conceitos abstratos de espaço geográfico”. Ou seja, cada disciplina possui seus próprios conceitos e definições. Sendo assim, é necessário traduzir estes conceitos em representações computacionais para que possam ser aplicados em Sistemas de Informação Geográficas (SIG).

Na construção de uma aplicação, um dos principais desafios é escolher as representações computacionais mais adequadas ao seu domínio de aplicação. Por isso, as tecnologias escolhidas para a construção da aplicação devem fornecer um amplo conjunto de estruturas de dados e algoritmos capazes de representar a grande diversidade de concepções do espaço.

Câmara [27] utiliza o "paradigma dos quatro universos" para explicar o processo de transformação de uma imagem do universo real em uma representação computacional. Os universos descritos pelo paradigma são:

a) O universo do mundo real: Inclui as entidades da realidade a serem modeladas no sistema, é onde se encontram os fenômenos a serem representados;

b) O universo matemático: Inclui uma definição matemática formal das entidades a serem representadas; pode-se distinguir entre as grandes classes formais de dados geográficos dados contínuos e objetos individualizáveis e especializar estas classes nos tipos de dados geográficos utilizados comumente;

c) O universo de representação: em que as diversas entidades formais são mapeadas para representações geométricas no computador; as entidades formais definidas no universo conceitual são associadas a diferentes representações geométricas, que podem variar conforme a escala e a projeção cartográfica escolhida e a época de aquisição do dado. Aqui se distingue entre as representações matricial e vetorial, que serão apresentadas adiante;

d) O universo de implementação: em que as estruturas de dados e algoritmos são escolhidas, baseados em considerações como desempenho, capacidade do equipamento e tamanho da massa de dados. É neste nível que acontece a codificação.

Neste capítulo, são abordados detalhes dos universos de representação e implementação, conceitos gerais de SIGs bem como algumas tecnologias web para geoprocessamento.

2.1 Universo de Representação

No universo de representação, definem-se as possíveis representações geométricas associadas às classes do universo conceitual e podem ser divididas em duas classes: a de representação matricial e a de representação vetorial. A escolha da representação apropriada dependerá do tipo de dado que será representado, podendo ser:

a) Dados Temáticos: Descrevem a distribuição espacial de uma grandeza geográfica de forma qualitativa e admitem tanto representação matricial quanto vetorial;

b) Dados Cadastrais: Possui objetos geográficos que contém atributos associados. Sua parte gráfica é armazenada em forma de coordenadas vetoriais e seus atributos não gráficos são guardados em um banco de dados;

c) Redes: Possui objetos gráficos com topologia arco-nó e cada elemento possui atributos associados. Sua parte gráfica é armazenada em forma de coordenadas vetoriais, com a topologia arco-nó e seus atributos não gráficos são guardados em um banco de dados;

d) Imagens de Sensoriamento Remoto: São imagens obtidas por satélites, fotografias aéreas ou "scanners" e são armazenadas em representação matricial;

e) Modelos Numéricos de Terreno: São modelos matemáticos que representam uma variação continua de uma superfície real. Podem ser armazenados em grades regulares (representação matricial), grades triangulares (representação vetorial com topologia arco-nó) ou isolinhas (representação vetorial sem topologia).

2.1.1 Representação Matricial

No modelo matricial, supõe-se que o terreno possa ser descrito como uma superfície plana. Dessa forma, é possível representá-lo como uma matriz M(i,j), em que cada célula, associada a uma porção do terreno possui uma coordenada (um número de linha e um número de coluna) e um valor referente ao atributo estudado. Quanto maior a relação entre o tamanho da célula e a área coberta por ela no terreno, maior será a resolução do sistema. Entretanto,

como consequência, será demandado mais espaço para o seu armazenamento. A Figura 2.1 apresenta dois exemplos de representações matriciais para mapas. A imagem da direita apresenta um tamanho de célula menor que a da esquerda, portanto uma maior resolução do terreno.

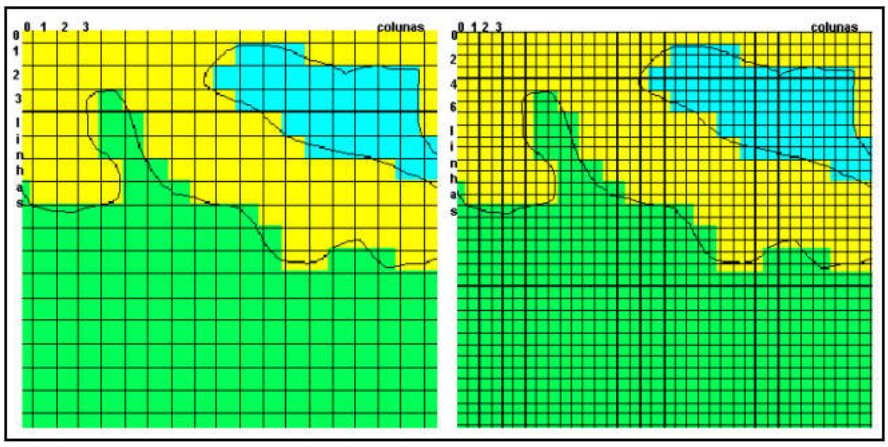


Figura 2.1 - Exemplos de representações matriciais para um mapa [27]

2.1.2 Representação Vetorial

Na representação vetorial são utilizados três elementos gráficos: ponto, linha, e polígono. Nela um ponto é um par ordenado (x,y), uma linha são um conjunto de pontos conectados e um polígono é uma região do plano delimitado por uma ou mais linhas conectadas de tal forma que o último ponto de uma linha seja o primeiro da próxima. Na criação da representação de um objeto, podem ser utilizados mais de um elemento primitivo básico. A Figura 2.2 apresenta os elementos primitivos básicos de uma representação vetorial.

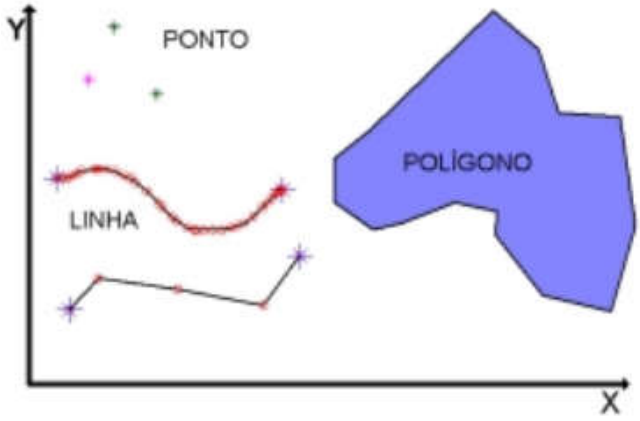


Figura 2.2 - Elementos primitivos de representação vetorial [27]

2.2 Universo de Implementação

Ao se discutir o universo de implementação, são indicadas quais as estruturas de dados utilizadas para construir um sistema de Geoprocessamento. Neste momento, são tratadas as decisões concretas de programação e que podem admitir um número muito grande de variações. Estas decisões podem levar em conta as aplicações às quais o sistema é voltado, a disponibilidade de algoritmos para tratamento de dados geográficos e o desempenho do hardware [27].

Um dos aspectos principais a ser levado em conta no universo de implementação é o uso de estruturas de indexação espacial. Os métodos de acesso a dados espaciais compõem-se de estruturas de dados e algoritmos de pesquisa e recuperação e representam um componente determinante no desempenho total do sistema. Estes métodos operam sobre chaves multidimensionais e dividem-se conforme a representação dos dados associados. A Tabela 2.1 apresenta algumas entidades do mundo real e suas correspondentes no universo conceitual, de representação e de implementação [27].

Tabela 2.1 - Correspondências entre universos do modelo

Universo Real	**Universo Conceitual**	**Universo de Representação**	**Universo de Implementação**
Mapa de vegetação	Geo-campo Temático	Matriz de inteiros Subdivisão Planar	Quad-tree Linhas 2D (com R-Tree)
Mapa altimétrico	Geo-Campo Numérico	Grade regular Grade triangular Conjunto Pontos 3D Conjuntos Isolinhas	Matriz 2D Linhas 2D e Nós 3D Pontos 3D (KD-Tree) Linhas 2D
Lotes urbanos	Geo-objetos	Polígonos e Tabela	Linhas 2D e Nós 2D
Rede elétrica	Rede	Grafo Orientado	Linhas 2D (com R-Tree)

2.3 Sistemas de Informação Geográficas

A terminologia Sistemas de Informação Geográfica (SIG) é aplicada para sistemas que realizam o tratamento computacional de dados geográficos e recuperam informações não apenas com base em suas características alfanuméricas, mas também por meio de sua localização espacial. SIGs oferecem ao usuário do software uma visão aprimorada de seu ambiente de trabalho, em que todas as informações disponíveis sobre um determinado assunto estão relacionadas à sua localização geográfica, que é a variável fundamental para o georreferenciamento. Para que isto seja possível, a geometria e os atributos dos dados num

SIG devem estar georreferenciados, isto é, localizados na superfície terrestre e representados em um sistema de coordenadas [27]. Existem três formas fundamentais de se usar um SIG:

a) como ferramenta para produção de mapas;

b) como suporte para análise espacial de fenômenos;

c) como um banco de dados geográficos, com funções de armazenamento e recuperação de informação espacial.

No âmbito do projeto, todas as três formas fundamentais de utilização serão abrangidas, como apresentado nos próximos capítulos deste trabalho. A estrutura geral de um SIG, de forma abrangente, pode conter os seguintes componentes:

a) interface com usuário;

b) entrada e integração de dados;

c) funções de consulta e análise espacial;

d) visualização e plotagem;

e) armazenamento e recuperação de dados organizados sob a forma de um banco de dados geográficos.

A Figura 2.3 ilustra a estrutura geral de um SIG. Estes componentes se relacionam de forma hierárquica. No nível mais próximo ao usuário, a interface homem-máquina define como o sistema é operado e controlado. No nível intermediário, um SIG deve ter mecanismos de processamento de dados espaciais (entrada, edição, análise, visualização e saída). No nível mais interno do sistema, um sistema de gerência de bancos de dados oferece armazenamento e recuperação dos dados espaciais e seus atributos. A integração entre os dados geográficos e as funções de processamento do SIG é feita por mecanismos de seleção e consulta que definem seleções sobre o conjunto de dados [27].

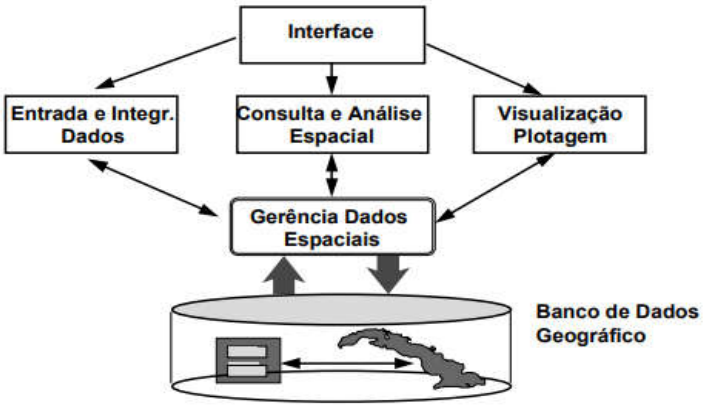


Figura 2.3 - Estrutura geral de um SIG [27]

Segundo Câmara, existem três arquiteturas de SIGs que utilizam os recursos de um Sistema Gerenciador de Banco de Dados (SGBD) [27]: a) arquitetura dual; b) arquitetura Integrada baseada em SGBDs Relacionais; c) arquitetura Integrada baseada em extensões espaciais sobre SGBDs objeto-relacionais. Elas são sintetizadas a seguir.

2.3.1 Arquitetura dual

Na arquitetura dual, utiliza-se um SGBD relacional para armazenar os atributos convencionais dos objetos geográficos e arquivos para guardar as representações geométricas destes objetos. No modelo relacional, os dados são organizados na forma de tabela onde as linhas correspondem aos dados e as colunas correspondem aos atributos. A entrada dos atributos não espaciais é feita por meio de um SGBD relacional e para cada entidade gráfica inserida no sistema é inserido um identificador único ou rótulo, por meio do qual é feita uma ligação lógica com seus respectivos atributos não espaciais armazenados em tabelas de dados no SGBD [27].

A principal vantagem desta estratégia é poder utilizar os SGBDs relacionais. No entanto, como as representações geométricas dos objetos espaciais estão fora do controle do SGBD, esta estrutura dificulta o equacionamento das questões de otimização de consultas, gerência de transações e controle de integridade e de concorrência [27].

2.3.2 Arquiteturas Integradas

A arquitetura Integrada, consiste em armazenar todo o dado espacial em um SGBD, tanto sua componente espacial como a parte alfanumérica. Sua principal vantagem é a utilização dos recursos de um SGBD para controle e manipulação de dados espaciais, como gerência de transações, controle de integridade e concorrência. Sendo assim, a manutenção de integridade entre a componente espacial e alfanumérica é feita pelo SGBD [27].

a) Baseada em SGBDs Relacionais

A arquitetura integrada baseada em um SGBD relacional utiliza campos longos, chamados de BLOBs (*Binary Large Objects*), para armazenar a componente espacial do dado, porém, não é capaz de capturar a semântica dos dados espaciais. Como o SGBD trata o campo longo como uma cadeia binária, não é possível conhecer a semântica do seu conteúdo. Além disso, métodos de acesso espacial e otimizador de consultas devem ser implementados pelo SIG, pois como o SGBD trata os dados espaciais como uma cadeia binária, não possui

mecanismos satisfatórios para o seu tratamento. Outra desvantagem são as limitações da linguagem SQL para a manipulação dos dados espaciais, a SQL padrão oferece recursos limitados para o tratamento de campos longos.

b) Baseada em extensões espaciais sobre SGBDs objeto-relacionais

O outro tipo de arquitetura integrada consiste em utilizar extensões espaciais desenvolvidas sobre SGBDs objeto-relacionais (SGBDOR). Estas extensões contêm funcionalidades e procedimentos que permitem armazenar, acessar e analisar dados espaciais de formato vetorial. Como desvantagens dessa arquitetura podem ser citadas as faltas de mecanismos de controle de integridade sobre os dados espaciais e a falta de padronização das extensões da linguagem SQL. Os SGBDs objeto-relacionais, também chamados de SGBDs extensíveis, oferecem recursos para a definição de novos tipos de dados e de novos métodos ou operadores para manipular esses tipos, estendendo assim seu modelo de dados e sua linguagem de consulta. Por isso, um SGBDOR é mais adequado para tratar dados complexos, como dados geográficos, do que um SGBDR, o qual não oferece esses recursos [27].

2.4 Tecnologias para Geoprocessamento

O Geoprocessamento é um ramo da área do conhecimento denominada oficialmente de Geomática. Ele engloba o total conjunto de técnicas ou tecnologias ligadas à informação espacial, quer seja no tocante a coleta, tratamento e análise desses dados. Algumas dessas técnicas, também chamadas de geotecnologias são Topografia, Fotogrametria, Cartografia, Sensoriamento Remoto, Posicionamento por Satélite, Geoestatística, Banco de Dados Geográficos, WebMapping e SIG [28].

A seguir, serão listadas algumas geotecnologias utilizadas no desenvolvimento deste projeto e do sistema SIGAEDES, discutido mais a seguir. Os principais critérios para escolha destas ferramentas decorreram do fato de serem tecnologias de código aberto e pela grande quantidade de material disponível para estudo das ferramentas.

Na Figura 2.4 é apresentado um diagrama com todas as tecnologias efetivamente utilizadas na construção do sistema. As atividades no bloco *client-side* são serviços executados na maquina do usuário do sistema, as do bloco *server-side* são aplicações executadas no servidor e as tecnologias fora dos blocos são utilizadas para pré processamento.

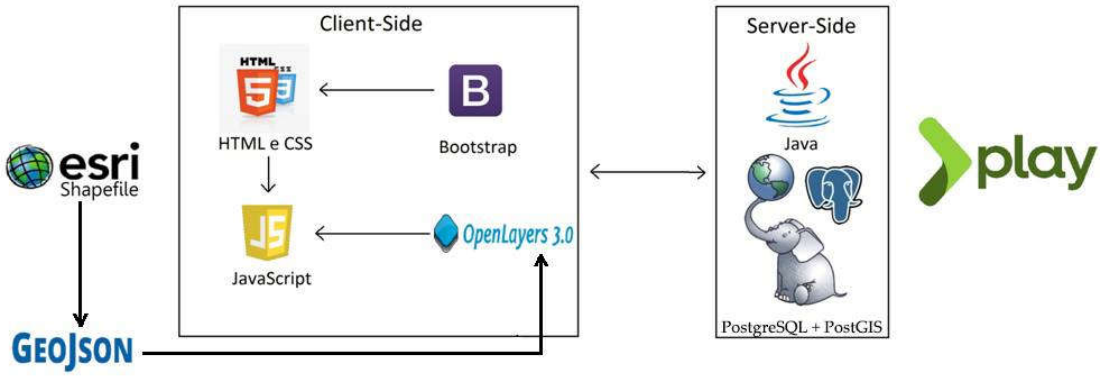


Figura 2.4 – Diagrama das tecnologias utilizadas na construção do SIGAEDES

2.4.1 QGIS

Quantum GIS (QGIS) [29] é um software de código aberto multiplataforma de sistema de georreferenciamento (SIG) que provê visualização, edição e análise de dados georreferenciados. Similar a outros softwares SIG, o QGIS permite ao usuário criar mapas com várias camadas usando diferentes projeções de mapa. Permite a criação de mapas em diferentes formatos e para diferentes usos. Também permite compor mapas a partir de camadas matriciais ou vetoriais.

2.4.2 PostgreSQL/PostGIS

O PostgreSQL [13] é um Sistema gerenciador de banco de dados objeto relacional (SGBDOR) de código aberto, desenvolvido atualmente pela *PostgreSQL Global Development Group. N*a necessidade de estender este SGBDOR para suportar dados espaciais, foi desenvolvida a extensão conhecida como PostGIS [14]. Portanto o PostGis é um *plugin* do PostgreSQL que adiciona ao SGBDOR funções para armazenamento e manipulação de dados geográficos.

Entre as funções adicionadas está a de importação de arquivos no formato *Shapefile* [30], para dentro de um Banco PostGIS utilizando recursos oferecidos pelo próprio programa ou utilizando algum software de SIG com essa funcionalidade. O shapefile pode ser convertido em uma tabela espacial que pode ser integrada com as convencionais contidas na base, além de poder ser visualizada e manipulada através de programas como QGIS [28].

2.4.3 Openlayers

A geoinformação, como área do conhecimento, também encontrou na internet um nicho para suas atividades. A disponibilização de mapas digitais online, os chamados WebGIS ou Webmapping, tem-se tornado comuns, permitindo que um maior número de usuários tenha acesso à dados espacializados [28].

Já existem vários softwares e bibliotecas para desenvolvimento de aplicações de Webmapping dentre eles encontra-se o Openlayers [18], que é uma biblioteca JavaScript [16] de código aberto, que permite exibição de dados espaciais em páginas web. Ele fornece uma API para construção de aplicações geográficas baseadas na web.

Capítulo 3

Monitoramento da Dengue no Brasil

Os principais sistemas utilizados pelo Ministério da Saúde para monitoramento da Dengue no Brasil são: Sistema de Informação de Agravos de Notificação (SINAN), Levantamento de Índice (LI), Levantamento Rápido do Índice de Infestação por Aedes aegypti (LIRAa), e o Sistema do Programa Nacional de Controle da Dengue (SisPNCD).

O SINAN agrupa informações relativas à notificação e investigação de casos de doenças e agravos que constam na Lista Nacional de Doenças de Notificação Compulsória, que inclui a dengue. O objetivo do SINAN é realizar a identificação dinâmica da ocorrência de um dado evento na população. Contribui como subsídio para melhor entendimento do agravo, além de indicar riscos aos quais os indivíduos estão sujeitos, caracterizando a situação epidemiológica de determinada área geográfica [31].

O Levantamento de Índice (LI) é um indicador amostral obtido através de pesquisa larvária que informa o grau de infestação, dispersão e densidade dos vetores em localidades do município. O Ministério da Saúde estabelece que todos os imóveis devem ser inspecionados e que sejam coletadas larvas ou pupas em 33% dos imóveis existentes na zona de trabalho. Assim, todos os imóveis são inspecionados, mas a coleta é realizada em um terço dos imóveis visitados. Em conjunto com o LI, são realizadas pesquisas entomológicas em pontos estratégicos e outras ações visando o monitoramento do *Aedes aegypti* [32]. Há um software específico de mesmo nome através do qual são inseridos os dados colhidos na ação do LI.

O levantamento Rápido do Índice de Infestação por *Aedes Aegypti* (LIRAa) é um sistema no qual são inseridos os dados coletados durante a ação de mapeamento rápido dos índices de infestação do mosquito. As capitais e municípios de regiões metropolitanas, municípios com mais de 100 mil habitantes e municípios com grande fluxo de turistas ou situados em regiões de fronteira devem realizar este levantamento. Para isso, o município é dividido em grupos de 9 a 12 mil imóveis com características semelhantes. Em cada grupo, são pesquisados 450 imóveis. Os grupos com índices de infestação predial inferiores a 1% são considerados como estando em condições satisfatórias. Aqueles entre 1% a 3,9% estão em situação de alerta. Considera-se que municípios com índices superiores a 4% há risco de surto de dengue [33]. Em síntese, o LIRAa aplica-se às situações em que se deseja avaliar o impacto de medidas de controle vetorial em áreas recém-infestadas ou como subsídio para avaliar os programas municipais.

O SisPNCD é o sistema através do qual são registradas as três principais ações realizadas em campo pelos agentes de controle de endemias: o Ciclo, o Ponto Estratégico (PE) e a Pesquisa Vetorial Espacial (PVE). O Ciclo consiste em visitas bimestrais aos imóveis do município, visando à orientação a moradores, retirada, destruição ou limpeza de depósitos que podem acumular água e aplicação de larvicida. Os PEs são locais estratégicos, a exemplo de depósitos de lixos, cemitérios, borracharias, que, por suas características, podem contribuir para a proliferação do vetor. Esses locais são vistoriados quinzenalmente com aplicação de larvicida e adulticida. O PVE, também chamado de Raio, é a ação realizada na circunvizinhança de 300 metros do local de maior permanência do indivíduo com suspeita de estar com a doença ou do local onde foram localizadas larvas ou pupas do vetor. Nesses locais é feita a remoção de depósitos e aplicação de larvicida.

Outras ações realizadas no SisPNCD são os cadastros de pontos estratégicos, de veículos que realizam aspersão aeroespacial de inseticidas em ultrabaixo-volume (UBV), armadilhas, áreas, microáreas e recursos humanos. O sistema emite relatórios como os de localidades, de pontos estratégicos, de indicadores entomológicos, de índice de recipientes, totais de produção e consumo de inseticida utilizado nas ações envolvendo aplicações focais, perifocais e de UBV. As aplicações focais consistem no uso de larvicida nos depósitos positivos para formas imaturas de mosquitos, que não possam ser eliminados mecanicamente. As perifocais consistem na aplicação de uma camada de inseticida de ação residual nas paredes externas dos depósitos em pontos estratégicos, através de aspersor manual, visando atingir o mosquito adulto que pousa para repouso ou desova. As aplicações de UBV são restritas às epidemias, visando à rápida interrupção da transmissão da doença. Este método atinge a superfície do corpo do mosquito e apresenta alto rendimento com maior área tratada por unidade de tempo [34].

A principal utilização dos dados coletados através destes sistemas tem sido a produção de dados estatísticos nacionais e, apesar de o preenchimento e envio para as regionais de saúde serem obrigatórios, os sistemas não são integrados e possuem limitações tanto na inserção como no fornecimento de informações para tomadas de decisões, forçando os gestores a trabalharem baseando-se em suas experiências e desperdiçando boa parte do potencial desta base de dados.

3.1 Sistemas de Informações no Âmbito da Dengue no Brasil e no Mundo

Além dos sistemas utilizados pelo Ministério da Saúde, existem outras iniciativas no âmbito da dengue no que tange a sistemas de informações e de modelagem e simulação em Dengue. Algumas são listadas a seguir.

a) DengueME: Elaborado pelo Departamento de Computação da Universidade Federal de Ouro Preto com a participação de diversos pesquisadores, é uma plataforma colaborativa de código aberto para simular a doença da dengue e sua dinâmica de vetores. Tem como objetivo representar a dinâmica da população de Aedes aegypti, a demografia humana, a mobilidade humana, a paisagem urbana e a transmissão da dengue baseada na interação de humanos e mosquitos. As simulações permitem a projeção de cenários e também avaliar a relação custo/benefício das diversas formas de controle da doença. Ele está sendo construído de maneira a ser de fácil manipulação para viabilizar que, além de pesquisadores acadêmicos, também profissionais de saúde possam realizar análises voltadas ao combate da dengue.

O módulo epidemiológico atual contempla um modelo baseado em agentes que atuam em um espaço celular acoplado a um Sistema de Informações Geográficas, para simulação e análise de cenários da dinâmica espaço-temporal da difusão da dengue em ambientes urbanos reais. O módulo entomológico permite, através de dois modelos, simular o ciclo de vida do *Aedes aegypti* bem como as estratégias de controle da dengue que tem como alvo o vetor [35];

b) Observatório Nacional da Dengue: O Projeto voltado ao Estado do Rio Grande do Norte está inserido dentro das ações do Projeto Telessaúde, que por usa vez é uma iniciativa instituída em âmbito nacional pelo Ministério da Saúde, desde 2007, para fortalecer e ampliar as ofertas de Educação Permanente em Saúde para os profissionais e trabalhadores do SUS utilizando tecnologia de comunicação e informação.

O Observatório Nacional da Dengue tem por objetivo monitorar e ajudar a população e o Gestor Público no Combate à Dengue por meio de um sistema que possibilite o monitoramento dos focos da dengue em tempo real e ofereça uma ferramenta que agilize o trabalho de campo do Agente de Endemias. A proposta é a de que as informações decorrentes das visitas realizadas pelos agentes de campo sejam informadas por meio de dispositivos móveis, fornecendo informações em tempo real sobre as visitas realizadas pelos agentes de campo no âmbito estadual, viabilizando que o cidadão possa ter acesso e acompanhar o trabalho realizado [36];

c) Observatório da Dengue: Desenvolvido pelo Instituto Nacional de Ciência e Tecnologia em Dengue, Instituto Nacional de Ciência e Tecnologia para a Web, em parceria com Ministério da Saúde e a Organização Pan-Americana de Saúde (OPAS). O Observatório da Dengue é capaz de coletar, analisar e apresentar em tempo real informações acerca da dengue a partir de mais de uma centena de fontes de dados internet, incluindo redes sociais e blogs, além de canais da mídia tradicional. O sistema permite visualizar as informações coletadas de diversas formas e provê estimativas acerca da incidência da dengue em determinada região em tempo real, sem o atraso que ocorre quando há a necessidade da notificação e entrada de dados no sistema tradicional de controle epidemiológico [37];

d) MI-Dengue: Software para dispositivos móveis desenvolvido pela empresa Ecovec [38] associado ao serviço de mesmo nome, que consiste no monitoramento de vetores adultos, que permite ao município obter um panorama semanal da infestação, utilizando um pequeno número de agentes de saúde. O MI-Dengue utiliza armadilhas desenvolvidas para captura de mosquitos Aedes. Elas são posicionadas dentro da área urbana do município e são vistoriadas semanalmente pelos agentes de endemias. O número de capturas em cada armadilha fornece as informações sobre a população do vetor. Essas informações são enviadas para a central de dados através de dispositivos móveis. As informações recebidas constituem a base de dados do site para gestão online, onde estão disponíveis os mapas de infestação do vetor, tabelas de incidência por bairros, relatórios de plano de ação para controle, além da área de gestão de equipe e insumos [38];

e) DengueNet: Desenvolvido pela Organização Mundial de Saúde (OMS) em 2005, é um sistema de gerenciamento central de dados para a vigilância epidemiológica e virológica mundial da dengue baseado na internet. Seu objetivo principal é disponibilizar uma plataforma padrão para o compartilhamento de dados visando detectar e monitorar a incidência e as tendências da dengue. O sistema viabiliza acesso a indicadores, como a incidência, letalidade, a freqüência e a distribuição, a quantidade de mortes e os sorotipos circulantes. Os dados são apresentados graficamente, em mapas e formatos tabulares e objetivam fornecer informações que possam ser úteis a profissionais de saúde pública e gestores nas ações contra a doença. Estatísticas sobre a dengue podem ser acessadas a partir de 1955 e seu foco atual está na coleta global de dados e no estabelecimento de parcerias [39];

f) Info Dengue: Sistema desenvolvido pela Escola de Matemática Aplicada da Fundação Getulio Vargas (FGV/EMAp) em parceria com a Fundação Oswaldo Cruz (FIOCRUZ) Secretaria Municipal de Saúde do Rio de Janeiro, a Universidade Federal de Minas Gerais (UFMG) e a Universidade Federal do Paraná (UFPR). O Info Dengue é um sistema de mapeamento de risco que utiliza dados coletados a partir de relatórios semanais das secretarias dos municípios com números de casos notificados por bairro e índices de infestação, além de indicadores climáticos. O principal diferencial do Info Dengue é a utilização de dados coletados em redes sociais como o Twitter, filtrando mensagens por palavras chaves específicas e por sua geolocalização e o mecanismo que utiliza para realizar alertas da doença [40]. Trata-se de um sistema cuja participação está aberta aos municípios interessados que a ele podem aderir fornecendo os referidos dados. A expectativa é de que por meio dele seja possível ampliar o conhecimento sobre a disseminação e o controle da dengue em âmbito nacional [41];

g) Google Dengue Trends: Ferramenta experimental desenvolvida pela Google analisava buscas relacionadas à Dengue no Brasil, Bolívia, Índia, Indonésia e Cingapura e buscava por correlações com os dados oficiais divulgados por estes países. A ferramenta realizava uma análise dos padrões de buscas entre os usuários para determinar se os períodos em que há mais pesquisas sobre dengue coincidem com o aumento no número de ocorrências da doença. Apesar de não serem mais publicadas estimativas atuais, as estimativas históricas continuam disponíveis no site do projeto [42];

h) Monitora Dengue: Software que tem como objetivo facilitar a captura de dados de visitas de agentes da dengue em campo no Rio de Janeiro. O software é utilizado tanto pelos agentes em campo quanto por seus supervisores, que distribuem e coletam os arquivos com os dados entre seu dispositivo móvel e os dos agentes. A informatização do registro dos agentes de vigilância em campo reduz o tempo de processamento das informações, melhora a qualidade das informações coletadas, georreferencia as visitas e geram índices e/ou indicadores com capacidade de análise na própria ferramenta. Esta possibilita ainda o planejamento e controle da execução das atividades da Vigilância em Saúde, tais como cadastro de usuários em vários níveis: Gestores, Supervisores e Agentes e planejamento das Atividades de Vigilância em Saúde e registro da verificação de Amostras em Laboratórios [43];

i) Framework Computacional da Dengue: É um conjunto de ferramentas computacionais para controle de doenças endêmicas tropicais com objetivo de acompanhar a evolução de focos e casos da dengue. Isto é feito por meio do tratamento de informações georreferenciadas e climatológicas. Dentre as ferramentas que constituem o framework estão o Geographvs que é um sistema desenvolvido em delphi que realiza a montagem, edição e correção de mapas com camadas de uma estrutura urbana (bairros, logradouros, quadras, imóveis, hidrografia, malha viária, e outras). O Sistema Gestor, que realiza a geração de escala, roteiros de visitas e acompanhamento destes roteiros, conforme a padronização de tarefas e divisão da carga de trabalho planejada pelo supervisor geral para os agentes sanitaristas e o Sistema WebDengue que possibilita a visualização de pontos estratégicos e quadras, acompanhamento de visitas, cruzamento de dados e região de influência de pontos estratégicos sobre casos e áreas de risco cruzando com informações populacionais do IBGE [44].

3.2 O Programa AEDES

O Programa AEDES [45], envolve ações de pesquisa e de extensão e atualmente esta sendo desenvolvido no Laboratório de Computação Aplicada a Saúde da UNIOESTE com a colaboração da Prefeitura Municipal de Cascavel. Seu objetivo é desenvolver um conjunto de atividades visando mitigar os problemas causados por essas doenças por meio de soluções efetivas à identificação de infestação, controle e combate vetorial de *Aedes Aegypti* e *Aedes Albopictus* [45].

Dentre as atividades realizadas no âmbito do programa, duas se destacam, aquelas relacionadas ao desenvolvimento dos módulos do Sistema de Informação Georreferenciada Aedes (SIGAEDES), que serão detalhados mais a frente neste texto, e aquelas relacionadas ao desenvolvimento do módulo de simulação em dengue. Neste módulo são feitas modelagens computacionais da dinâmica do espalhamento da dengue, empregando diversas abordagens, como redes de contato, autômatos celulares, agentes computacionais entre outras, objetivando construir possíveis cenários para que o gestor tome medidas preventivas controlando situações determinantes à ocorrência da epidemia.

3.3 O Sistema de Informações - SIGAEDES

O Sistema de Informação para Aquisição, Manipulação e Tratamento de Dados sobre a Dengue (SIGDENGUE), foi concebido em 2012 a partir do trabalho de uma equipe multidisciplinar com integrantes acadêmicos, gerentes e funcionários ligados ao Setor de

Endemias de Cascavel. Seu principal foco foi acompanhar os trabalhos realizados pelo Setor quando da notificação de casos suspeitos ou confirmados de dengue, informações sobre pontos estratégicos, localidades, resultados laboratoriais decorrentes de material colhido em visitas a campo, equipes de trabalho, entre outras caracterizações. Objetivava também integrar os dados dos sistemas SINAN, LI, LIRAa e SISFAD/SisPNCD, disponibilizados pelo governo federal, e a partir deles gerar informações úteis aos agentes do setor, capazes de subsidiar as decisões e as ações decorrentes no que se refere à dengue [45]

Considerando as funcionalidades disponibilizadas pelo SIGDENGUE, diversas foram as ocasiões de contato com representantes do Setor de Endemias, usuários do sistema, visando identificar melhorias e novas funcionalidades. Melhorias foram sendo implementadas diretamente no sistema objetivando seu aperfeiçoamento e melhor adaptação ao trabalho realizado por aquele setor.

Paralelamente, novas funcionalidades foram identificadas e consideradas como fundamentais para comporem uma nova versão do sistema, entre elas estavam a viabilização da utilização do SIGDENGUE por meio da internet, a possibilidade de realização de funcionalidades de georreferenciamento e acompanhamento dos casos de Chikungunya e Zika.

A necessidade de funcionalidades de georreferenciamento se justifica pelo fato de que atualmente o trabalho de combate e prevenção dos vetores de Dengue, Zika e Chikungunya realizado pelo Controle de Endemias de Cascavel possui como suporte apenas formulários em papel e croquis desenhados e atualizados a mão. Estes croquis possuem informações dos lotes de toda a cidade de Cascavel e servem como base para o planejamento de algumas ações operacionais de campo das equipes.

Conforme já mencionado, segundo o censo demográfico de 2010 realizado pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o município de Cascavel possui mais de 91 mil domicílios particulares ocupados [8]. Considerando que os croquis gerados pelos agendes do Controle de Endemias englobam todos os domicílios, sejam ocupados ou não, construções comerciais, terrenos baldios, entre outras categorias de terrenos, manterem todas essas informações atualizadas através de trabalho manual têm se tornado praticamente inviável.

Neste sentido as funcionalidades de consultas e visualizações de mapas se tornam uma parte essencial do sistema, se centrando na facilidade de visualização de resultados. Além disso, viabilizam sua associação com as áreas geográficas de ocorrência, oferecendo uma nova dimensão das relações que estabelecem, melhorando a qualidade das informações

armazenadas, das decisões tomadas pelos supervisores e minimizando o desperdício de tempo de trabalho manual dos agentes, para que possa ser empregado em ações mais importantes.

Durante o desenvolvimento da nova versão do sistema achou-se pertinente a alteração do nome SIGDENGUE para SIGAEDES, visto que o sistema não englobaria mais apenas os dados referentes aos casos de dengue, mas também de Zika e Chikungunya, cujas doenças são transmitidas pelo mesmo vetor, o *Aedes.*

Para uma melhor visualização das funcionalidades englobadas pelo SIGAEDES, na tabela 3.1 é apresentado um comparativo entre as funcionalidades previstas para o SIGAEDES, e as funcionalidades do SISFAD, SisPNCD, LI, LIRAa e SINAN, que são sistemas de utilização obrigatória, conforme o Ministério da Saúde, e o SIGDENGUE.

Tabela 3.1 – Comparação das funcionalidades disponibilizadas no SIGAEDES e demais sistemas utilizados pelo Setor de Endemias.

CARACTERÍSTICAS	SIGAEDES	SIGDENGUE	SISPNCD	LI	LIRAa	SINAN
Dados sobre pontos estratégicos	✓	✓	✓			
Dados sobre localidades	✓	✓	✓			
Dados sobre larvas e pupas	✓	✓	✓			
Dados sobre controle mecânico	✓	✓	✓			
Dados sobre controle químico	✓	✓	✓			
Levantamento de Índice de Infestação Predial	✓			✓		
Levantamento de Índice de Breteau	✓			✓		
Levantamento Amostral Instantâneo	✓	✓			✓	
Detalhamento de indivíduos com suspeita de dengue	✓	✓				✓
Identificação georreferenciada de indivíduos com suspeita de dengue	✓					
Acompanhamento do Raio associado ao caso suspeito	✓	✓				
Acompanhamento do Bloqueio associado ao caso confirmado	✓	✓				
Identificação dos casos suspeitos e confirmados de dengue	✓	✓				
Cadastro das equipes de campo	✓	✓	✓			
Dados sobre trabalho realizado pelas equipes de campo	✓	✓	✓			
Outras ações em dengue – mutirões	✓	✓				
Dados históricos	✓	✓	✓			

Apoio na tomada de decisão	✓	✓	✓			
Disponibilidade Web	✓					
Consultas e Visualização em Mapas	✓					
Identificação de casos suspeitos e confirmados de Zika	✓					
Identificação de casos suspeitos e confirmados de Chikungunya	✓					

Capítulo 4

O Geoprocessamento no SIGAEDES

Tendo em vista que um dos principais objetivos do SIGAEDES é integrar as atividades operacionais realizadas pelo Setor de Endemias de Cascavel, neste capítulo são detalhadas as atividades que demandam geoprocessamento, detalhes da implementação serão apresentados no capítulo 5.

4.1 Metodologia Operacional

A atividade de Reconhecimento Geográfico (RG) é uma ação desenvolvida constantemente pelos agentes do Setor de Controle de Endemias. Ela consiste em mapear os imóveis da cidade em croquis desenhados e atualizados a mão. Esses croquis possuem informações dos lotes de toda a cidade, como informações sobre logradouro, número, localidade, tipo de utilização, entre outras, e servem como base para o planejamento das ações operacionais de campo das equipes, como será visto mais a frente.

Esta se tornou a primeira demanda em termos de geoprocessamento para o SIGAEDES. Visto que todas as outras atividades operacionais se baseiam nos resultados desta atividade, viabilizar que o RG constasse diretamente no sistema de informações consiste em uma melhoria qualitativa significativa para aquele setor, o formulário de campo do Boletim de Reconhecimento pode ser visto no Apêndice B.

A partir do RG, os agentes definem os chamados Pontos Estratégicos (PE), que são pontos críticos na cidade como lotes baldios, borracharias ou cemitérios que necessitam ser vistoriados regularmente por serem imóveis com alta probabilidade de possuírem criadouros de mosquitos Aedes. Os PEs são visitados por equipes de campo a cada 15 dias procurando por focos de reprodução de mosquitos.

Outra atividade operacional importante, desenvolvida pelos agentes, é o rastreamento de casos suspeitos de Dengue, Zika e Chikungunya. O protocolo desta atividade inicia-se quando alguém procura uma unidade de saúde possuindo sintomas que indique alguma destas doenças. Ao ser atendido, é solicitado o preenchimento da Ficha de Informações de Agravo de Notificação que, dentre outras informações, solicita ao individuo que informe quais são seus endereços mais frequentados como endereço domiciliar, de trabalho, estudo, lazer, entre outros, estes endereços são regiões estratégicas para as ações seguintes.

Os formulários preenchidos nas Unidades de Saúde são encaminhados para o Setor de Controle de Endemias, onde os supervisores gerais analisam os casos suspeitos e dão inicio a atividade de Raio. Esta atividade consiste em uma ação de busca por criadouros de mosquitos *Aedes* em todos os lotes dentro de um raio de 150 metros ao redor de cada endereço informado pelo individuo com suspeita de Dengue, Zika ou Chikungunya. Para cada Endereço informado, é gerado um Raio e para cada Raio, uma equipe de agentes de campo é designada para efetuar a busca. As equipes são designadas de acordo com a região da cidade que o Raio pertence, e a prioridade de realização das ações é definida empiricamente pelos supervisores gerais baseando-se em proximidade com PEs ou com outros casos suspeitos ou confirmados.

Durante a realização das buscas, as equipes de campo vistoriam todos os lotes dentro do Raio do endereço do suspeito, coletando amostras de água e as categorizando de acordo com o tipo de depósito onde foram encontradas. As amostras são enviadas para o Setor de Entomologia do Controle de Endemias, onde são realizados testes para a confirmação de larvas ou pupas do mosquito Aedes.

Nos casos em que os exames realizados no indivíduo possuem resultados positivos pra Dengue Zika ou Chikungunya, ou os testes entomológicos apontam que foram encontradas larvas ou pupas do mosquito Aedes em alguma amostra coletada durante o Raio, imediatamente é deflagrada a ação de Bloqueio. Esta ação será realizada por uma equipe especializada em aplicação de veneno e o faz na região onde o caso da doença foi confirmado, com o objetivo de reduzir o potencial de ação do mosquito já adulto.

Ações de busca por criadouros também podem ser iniciadas através de denuncias da população ou dos próprios agentes de campo. Nestes casos o proprietário do terreno recebe uma notificação ou até mesmo uma multa, caso não realize a limpeza do lote dentro do prazo estipulado.

4.2 Prototipação das Funcionalidades Geoespaciais do SIGAEDES

Devido ao fato de as metodologias de trabalho dos agentes do Setor de Endemias serem tradicionais e aplicadas da mesma maneira durante anos, uma certa dificuldade foi encontrada durante o levantamento de requisitos das funcionalidades geoespaciais. A atividade de prototipação foi a forma encontrada para superar este obstáculo. Protótipos não funcionais foram desenvolvidos utilizando a ferramenta Pencil Project [46] por propiciar a criação de

protótipos rápidos e de fácil alteração, e assim, aumentando a qualidade das informações coletadas com os *stakeholders*.

A Figura 4.1 apresenta o protótipo da funcionalidade de cadastro do Reconhecimento Geográfico. Esta funcionalidade deve fornecer ao agente a visualização de um mapa com os lotes da cidade, viabilizando através do clique no lote à associação de informações ao mesmo. Quando o agente clica em um lote, o formulário a esquerda é automaticamente preenchido com as informações básicas do lote (logradouro, número), estas informações estão contidas diretamente no arquivo Shapefile fornecido pela prefeitura e inserido no banco de dados. Em seguida o agente pode adicionar as informações pertinentes ao Setor de Endemias como a localidade, quarteirão, sequencial, lado, número secundário, complemento, tipo de imóvel e observações a respeito do lote.

A Figura 4.2 apresenta o protótipo da funcionalidade de visualização dos endereços dos casos suspeitos. Esta funcionalidade deve viabilizar ao agente visualizar em mapa os vários endereços do suspeito selecionado bem como todas as informações a respeito dos endereços do suspeito.

A Figura 4.3 apresenta o protótipo da funcionalidade de geração de raios. Esta funcionalidade deve sugerir ao agente um conjunto de lotes a uma distância de até 150 metros ao redor do endereço do suspeito que deverão ser vistoriados por uma equipe de campo. Esta equipe de campo bem como as amostras coletadas durante esta ação também devem ser associadas a este raio nesta funcionalidade.

O protótipo da funcionalidade de visualização de Bloqueio Individual possui uma interface muito parecida com a de visualização do Raio Individual, pois utiliza o conjunto de lotes gerados pelo Raio como base para a ação de bloqueio. Assim como na funcionalidade de Raio, o agente tem acesso aos dados relativos ao suspeito que originou a ação e a funcionalidade de associação de agentes para realizar o bloqueio, porém não possui a função de associação de amostras, visto que não são coletadas amostras durante o Bloqueio.

O protótipo da funcionalidade de visualização de Pontos Estratégicos é apresentado na Figura 4.4. Nesta funcionalidade o agente deve ter a possibilidade visualizar informações sobre um ponto estratégico em específico e sua localização em mapa, bem como realizar o planejamento de visitas associando agentes de campo às visitas agendadas.

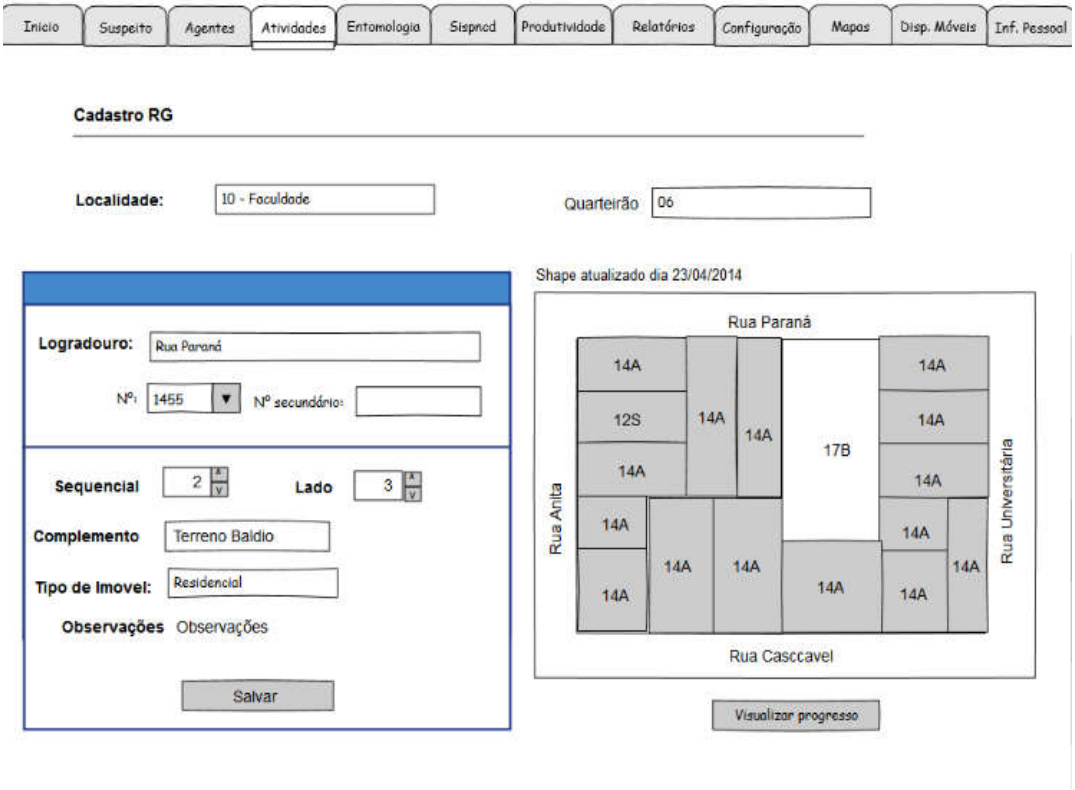


Figura 4.1 – Protótipo da funcionalidade Reconhecimento Geográfico.

A funcionalidade Tela do Cidadão foi uma proposta que surgiu durante as reuniões de especificação de requisitos. Ela não é associada com nenhuma atividade operacional já realizada pelo Setor de Endemias e tem como objetivo ser uma sessão do Sistema dedicada a informar a população sobre a situação atual da Dengue no município. A Figura 4.5 apresenta o protótipo desenvolvido para esta funcionalidade. Esta ferramenta deve apresentar números de casos confirmados e suspeitos de Dengue, de amostras positivas e de Pontos Estratégicos agrupados por localidade do município em um mapa, com o cuidado de não divulgar casos individuais ou informações pessoais dos suspeitos, bem como materiais educativos sobre a dengue e uma sessão para contato.

Os requisitos apresentados neste capítulo são aqueles selecionados para serem desenvolvidos e descritos no decorrer deste trabalho. Conforme será apresentado nos capítulos seguintes, durante a implementação, foi detectada a necessidade de algumas alterações nos modelos iniciais destes requisitos.

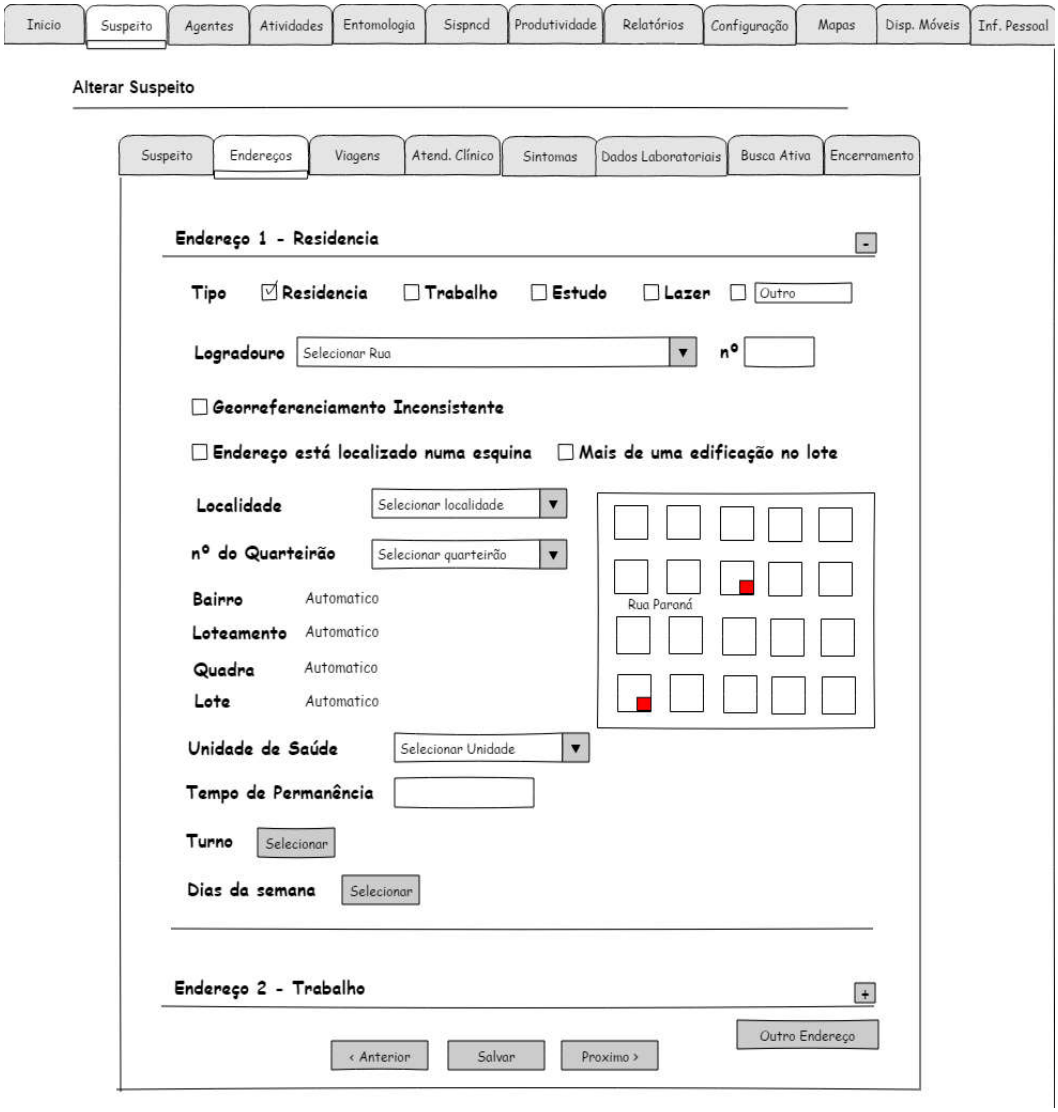


Figura 4.2 – Protótipo da funcionalidade Visualização de Caso Suspeito.

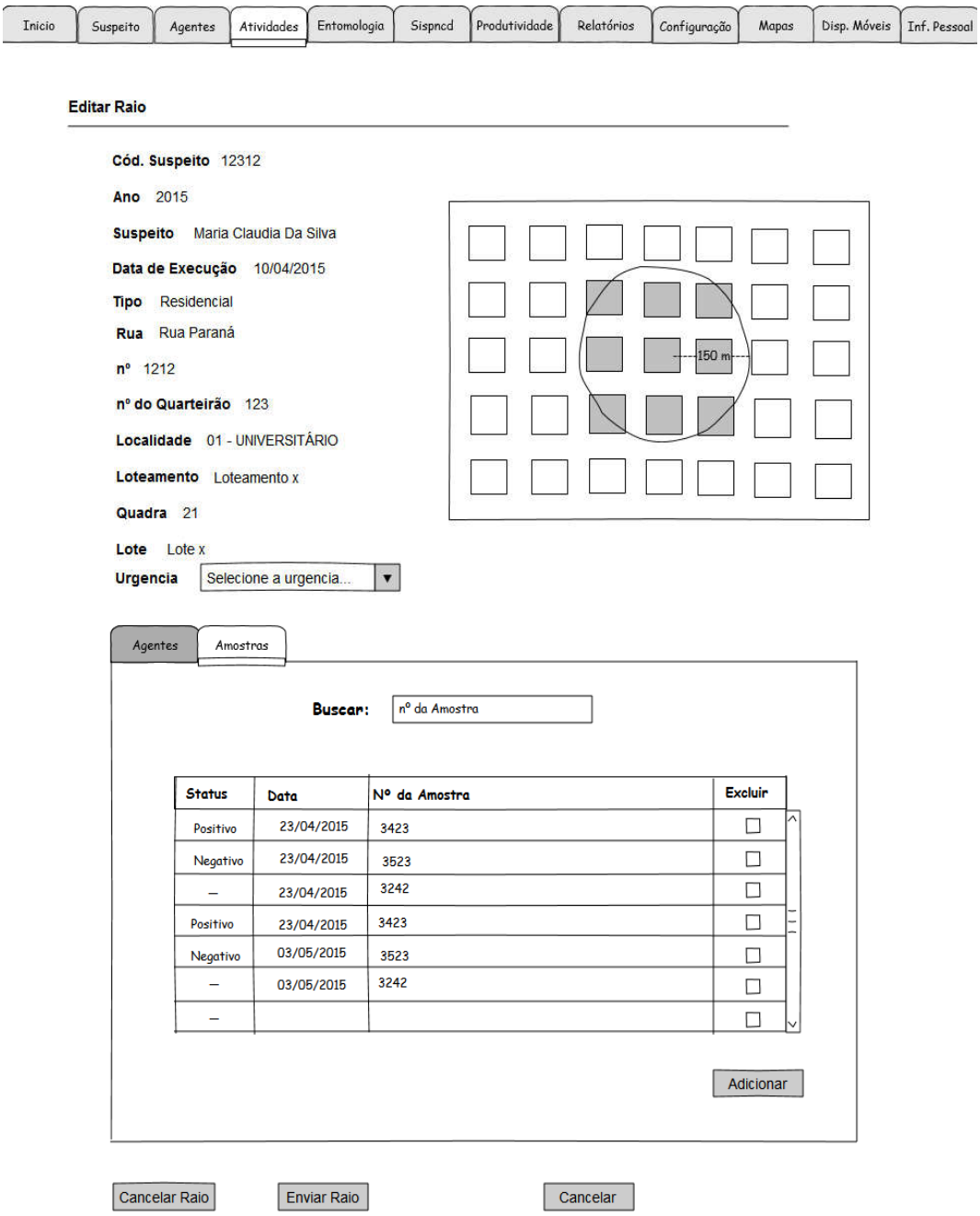


Figura 4.3 – Protótipo da funcionalidade Visualização de Raio Individual.

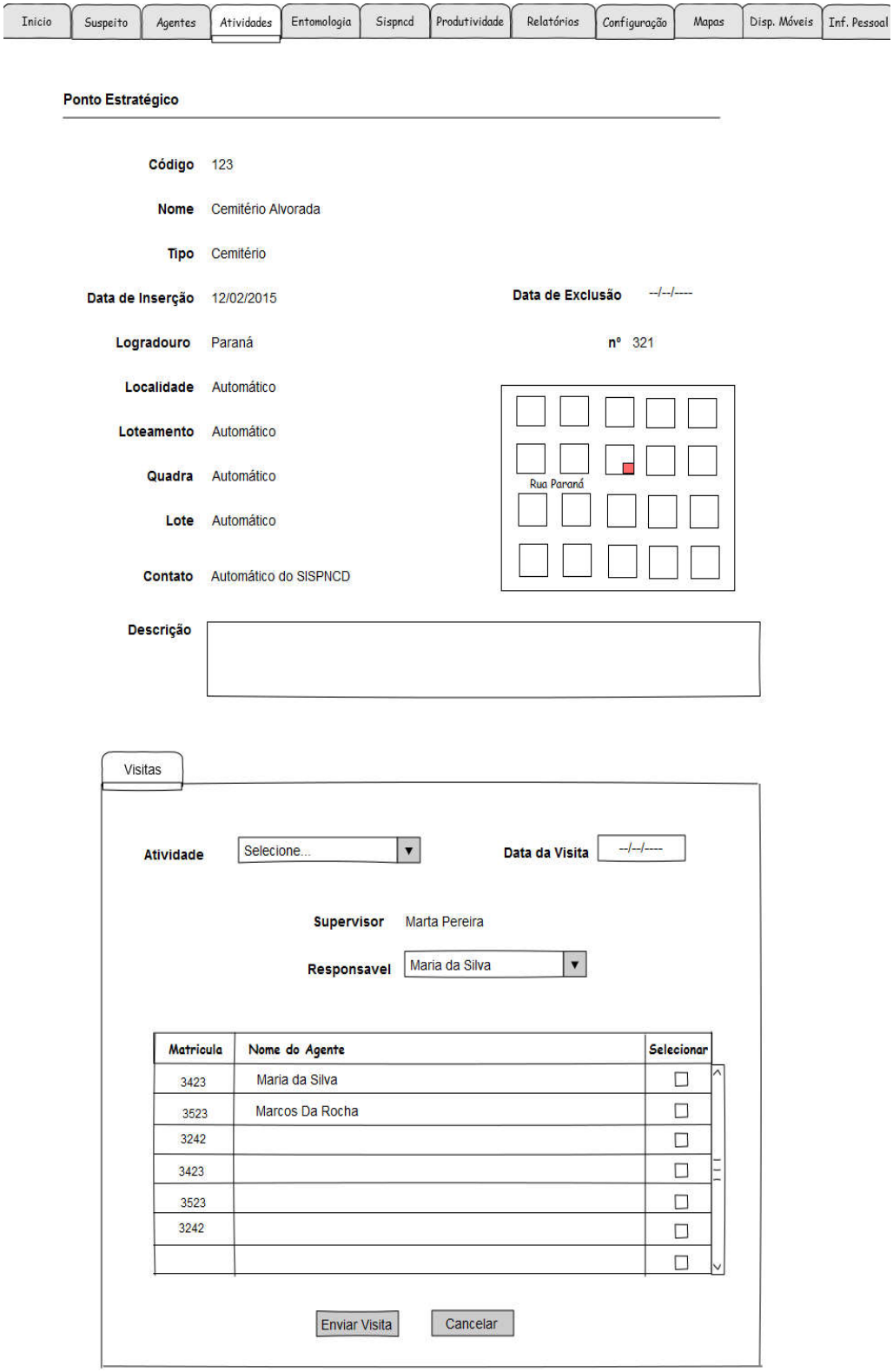


Figura 4.4 – Protótipo da funcionalidade Visualização de Ponto Estratégico.

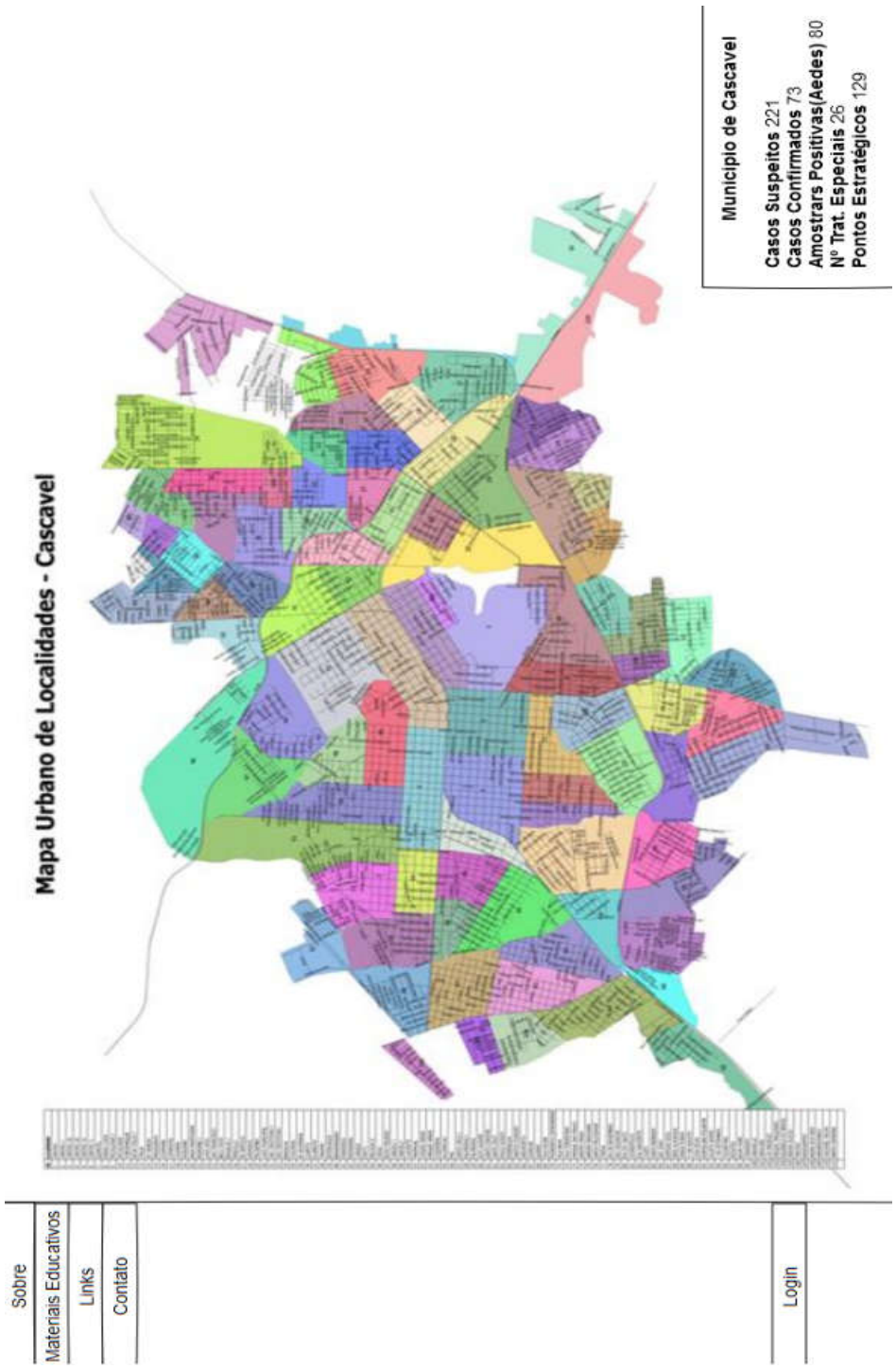


Figura 4.5 – Protótipo da funcionalidade Tela do Cidadão.

Capítulo 5

A Implementação do SIGAEDES

Neste capítulo são apresentados detalhes sobre a implementação das funcionalidades geoespaciais presentes no SIGAEDES e desenvolvidas durante a execução deste trabalho.

5.1 Arquitetura do Sistema de informações

Para garantir a reusabilidade, manutenção e organização do código do SIGAEDES foi utilizado o padrão de projeto Modelo-Visão-Controlador (MVC) que consiste em dividir o código da aplicação em três camadas que trocam informações. A camada Modelo é onde se encontram os dados da aplicação e as regras de negócios, é nesta camada que é definido como os dados vão estar dispostos no banco de dados. A camada Visão é a camada que interage diretamente com o usuário, apresentando as informações processadas pelo sistema e capturando os comandos inseridos. Por fim, a camada Controlador é onde os comandos inseridos pelo usuário serão convertidos em dados para utilização na camada Modelo, e os dados retornados da camada Modelo são transformados em informação para serem apresentados na camada Visão. A Figura 5.1 apresenta um diagrama do funcionamento do padrão de projeto MVC.

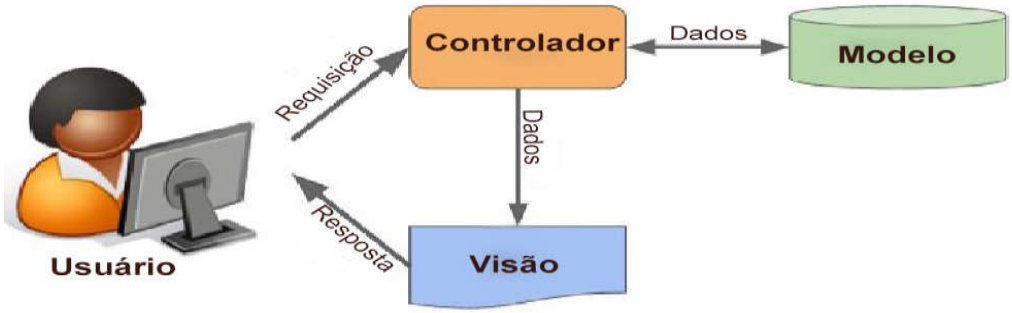


Figura 5.1 - Diagrama de funcionamento do padrão MVC. Adaptado de Helloacm [47].

Este trabalho tem como objetivo descrever as atividades desenvolvidas pelo autor durante o desenvolvimento das funcionalidades geoespaciais do SIGAEDES e não da implementação do sistema de forma geral, portanto os diagramas e algoritmos apresentados neste capítulo

terão foco nas atividades de apresentação de conteúdo geoespacial e as tabelas do banco de dados estarão disponíveis no Apêndice A a título de informação.

5.2 Arquiteturas das Funcionalidades Geoespaciais

Para as funcionalidades Tela do Cidadão e Reconhecimento Geográfico (RG) optou-se por utilizar uma arquitetura Dual apresentados no capítulo 2, pois os mapas que precisam ser apresentados para o usuário são sempre os mesmos em relação aos polígonos que os compõem.

Durante a etapa de prototipação da Tela do Cidadão previa-se que o mapa exibido nesta tela seria o mapa urbano de localidades, porém durante o desenvolvimento notou-se que um mapa dos bairros da cidade seria mais útil para o cidadão, visto que a noção de localidades é de utilização interna do Setor de Controle de Endemias e pouco conhecida fora deste setor. A Figura 5.2 apresenta a funcionalidade Tela do Cidadão mostrando os bairros de Cascavel com cores que representam a quantidade de casos do bairro em função da quantidade de indivíduos por bairro.

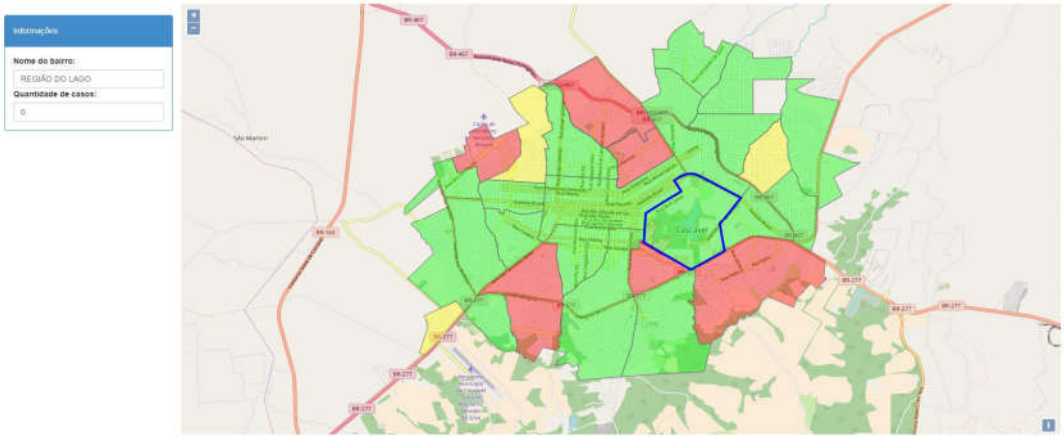


Figura 5.2 – Funcionalidade Tela do Cidadão.

A Figura 5.3 apresenta a lógica na implementação da funcionalidade, em seguida na Figura 5.4 pode-se observar o Diagrama de Entidade-Relacionamento das tabelas do banco de dados utilizadas.

Tela do cidadão
Controlador
Inicio
Para cada *bairro* **faça**:
Insere o numero total de habitantes do bairro em *total de habitantes*
Fim

Busca todos os *casos confirmados pertencentes a cidade*
Para cada *caso confirmado pertencente a cidade* **faça**:
Busca *endereço residencial do suspeito*
Busca *bairro do endereço residencial*
Incrementa o array *numero de casos por bairros* na posição do *codigo do bairro*
Fim

Insere *total de habitantes* na primeira posição da lista *estatisticas por bairros*
Insere *casos por bairros* na segunda posição da lista *estatistica por Bairros*;
fim

View
Inicio
Recebe a lista *estatistica por Bairros*
Para cada *bairro* **faça**:
Se *numero de casos do bairro* < 100 casos para cada 100.000 habitantes, **então**
Pinte o bairro de verde;
Se *numero de casos do bairro* > 100 casos para cada 100.000 habitantes < 300, **então**:
Pinte o bairro de amarelo;

Se *numero de casos do bairro* > 300 casos para cada 100.000 habitantes, **então**
Pinte o bairro de vermelho;
fim

Figura 5.3 – Logica na implementação da Tela do cidadão.

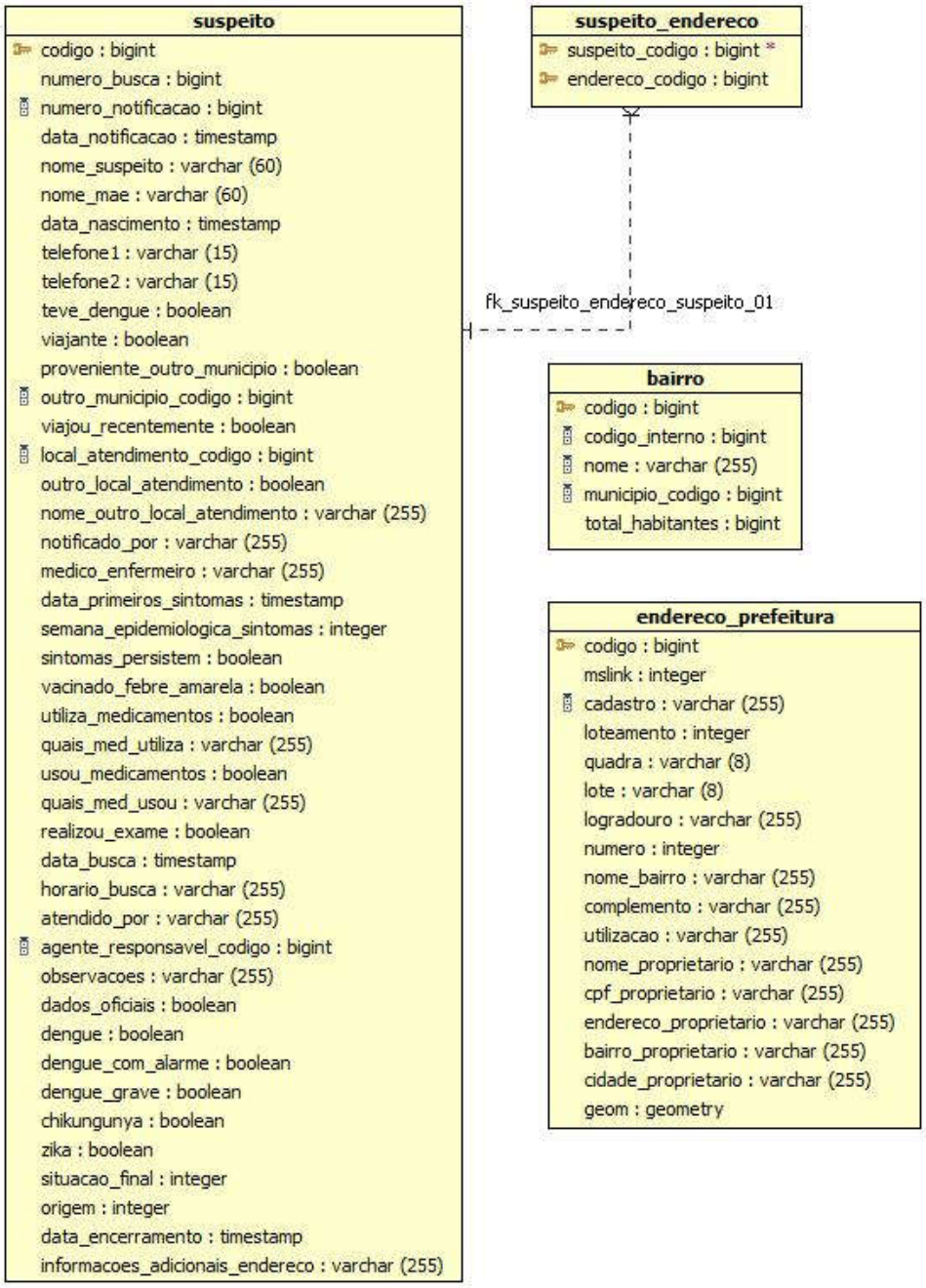


Figura 5.4 – Diagrama de Entidade-Relacionamento da funcionalidade Tela do Cidadão

O RG também sofreu alterações de requisitos a partir do protótipo. Nesta funcionalidade seria apresentado o mapa de lotes da cidade para cadastro e atualização de informações dos lotes e relacionamento entre as informações utilizadas pela Prefeitura Municipal de Cascavel e as informações utilizadas pelo Setor de Controle de Endemias. Porém, durante a implementação, percebeu-se que, mesmo utilizando a arquitetura dual, a

renderização de todos os polígonos presentes no arquivo era lenta demais devido à quantidade de polígonos, causando uma experiência ruim para o usuário, no sentido de tempo de espera para processamento.

A solução para corrigir este problema foi a de subdividir o mapa de lotes em vários arquivos de lotes por bairros para diminuir o número de polígonos renderizados por vez. Assim, o usuário passa a ter que selecionar qual bairro vai realizar o cadastro ou atualização para visualizar os lotes. A Figura 5.5 apresenta como ficou a tela de Reconhecimento Geográfico. A Figura 5.6 apresenta a lógica na implementação da funcionalidade em que o usuário clica em um lote do mapa eu as informações já cadastradas sobre o lote são carregadas para o formulário em tela e a Figura 5.7 apresenta o Diagrama de Entidade-Relacionamento desta funcionalidade.

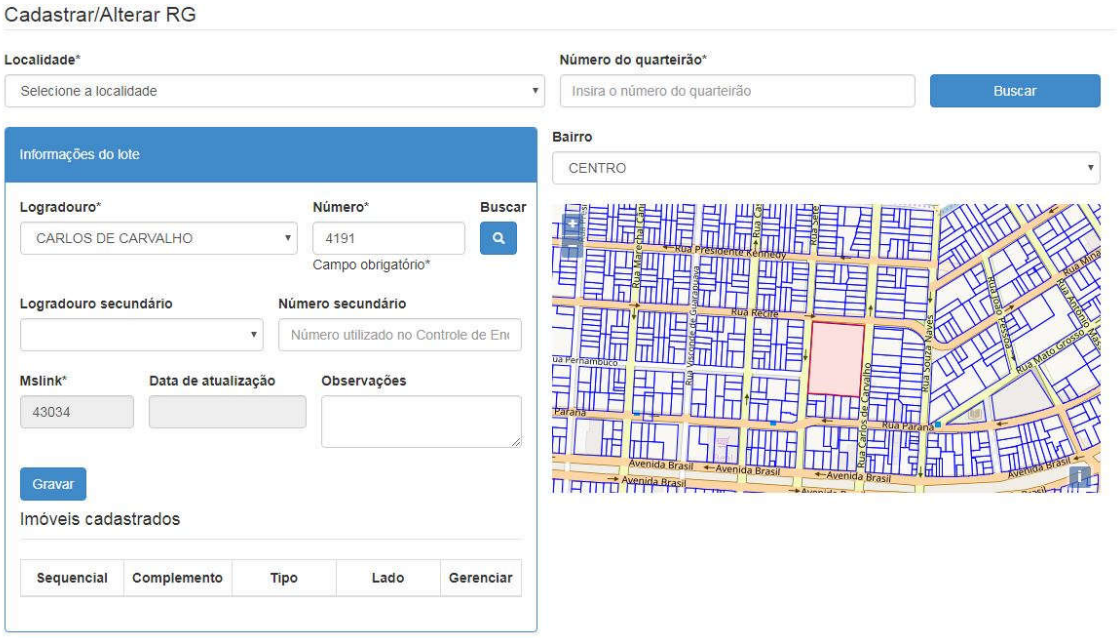


Figura 5.5 – Funcionalidade Cadastrar/Alterar RG.

Para as demais funcionalidades foi empregada a arquitetura Integrada baseada em extensões espaciais para sistemas gerenciadores de banco de dados objeto-relacionais (SGBDOR), devido à necessidade de mapas dinâmicos. O SGBDOR utilizado foi o PostgreSQL [13] juntamente com sua extensão espacial PostGIS [14], detalhes da implementação são apresentados no item seguinte.

```
Cadastro de RG
view
Inicio
        Se usuario clicou no mapa, então
                Captura coordenada do pixel clicado
                Captura lote exibido no pixel clicado
                Captura mslink do lote
        Fim
        Retorna mslink do lote
Fim

Controlador
Inicio
        Recebe mslink do lote clicado
        Busca informações do lote clicado
        Retorna informações do lote clicado
Fim
```

Pseudocódigo 5.6 – Lógica na implementação da funcionalidade de cadastro do RG, teclando no lote apresentado no mapa.

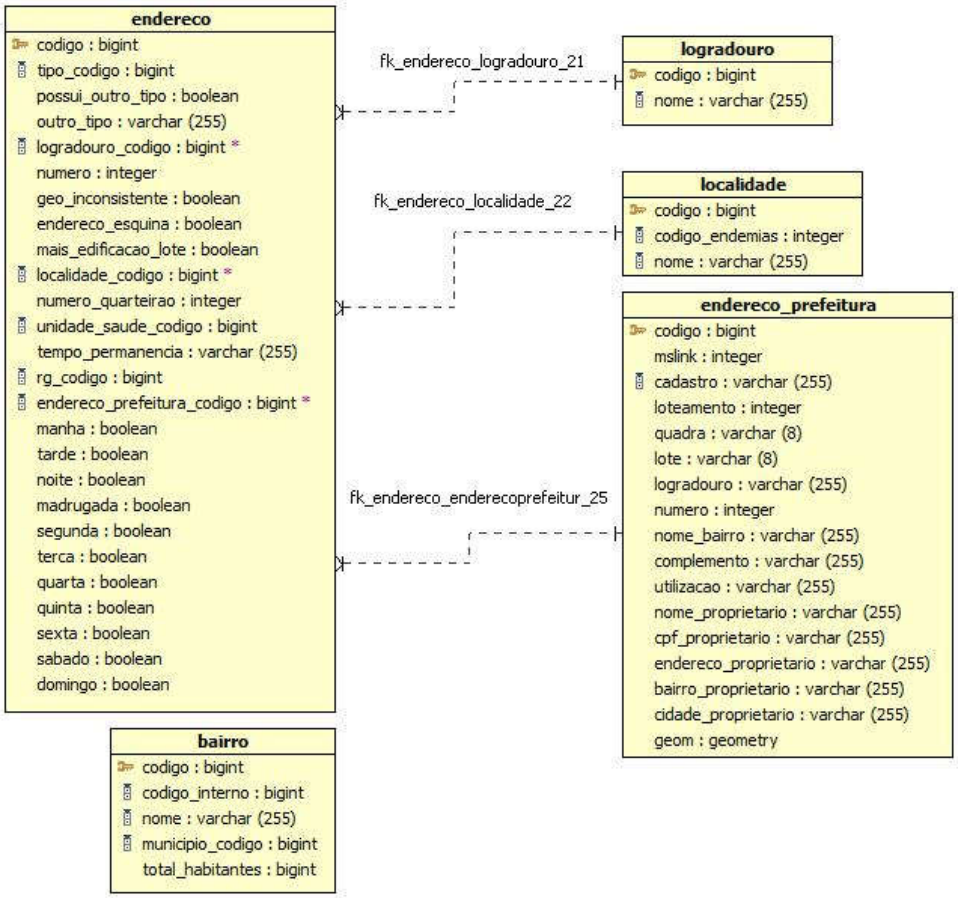


Figura 5.7- Diagrama de Entidade-Relacionamento da funcionalidade Cadastro de RG.

5.3 Implementação das Funcionalidades Geoespaciais

Para a implementação das funcionalidades que utilizam arquitetura Dual foi utilizado um arquivo no formato GeoJson [48] contendo as informações de geometria e um identificador para cada polígono. Para o arquivo contendo os bairros de Cascavel foi utilizado o código do bairro como identificador, já para o arquivo com os lotes por bairro foi utilizado o *mslink*, propriedade empregada pela prefeitura para individualizar cada lote da cidade, portanto cada lote possui um número de *mslink* único. Utilizando este identificador associado a cada polígono, é possível relacionar os atributos no banco de dados associados a esse polígono. A Figura 5.8 mostra o arquivo GeoJson que contém as informações de código e geometria do bairro Centro de Cascavel.

A funcionalidade de Visualização de endereços dos casos suspeitos é apresentada na Figura 5.9 e a Detalhar ponto Estratégico na Figura 5.10. Estas duas funcionalidades possuem uma implementação parecida como pode ser visto nos pseudocódigos apresentados nas Figuras 5.11 e 5.12. Esta semelhança deve-se ao fato de ambas consistirem em apresentação de marcadores no mapa representando informações pontuais. Em ambas as figuras, uma tarja preta foi inserida nos campos em que informações que podem ser consideradas pessoais ou privativas são apresentadas, objetivando atender a questões de sigilo e ética. Os respectivos Diagramas de entidade e Relacionamento das funcionalidades são exibidos nas Figuras 5.13 e 5.14.

```
{"type": "FeatureCollection", "crs": {
    "type": "name", "properties": {
        "name": "urn:ogc:def:crs:OGC:1.3:CRS84" } },
    "features": [
    { "type": "Feature", "properties": { "cdbairro": 1},
    "geometry":
        { "type": "Polygon", "coordinates": [ [ [ -53.480408966589536, -24.956370114790595 ],
            [ -53.480401750169506, -24.956267936376275 ], [ -53.479889225589211, -24.949019656568833 ],
            [ -53.479659871910612, -24.948596430270726 ], [ -53.479698847861343, -24.946274671624312 ],
            [ -53.469721027602873, -24.946849501284358 ], [ -53.459883624695372, -24.947417538311036 ],
            [ -53.458515592914765, -24.947487955530239 ], [ -53.458235562073931, -24.947502368110325 ],
            [ -53.45823999751034, -24.947526458931083 ], [ -53.458258609626078, -24.947566818640496 ],
            [ -53.458262716159638, -24.947589291923759 ], [ -53.458266823909312, -24.947611767172532 ],
            [ -53.458282272752385, -24.94763181853833 ], [ -53.458297820533588, -24.947651868300145 ],
            [ -53.458308818095361, -24.947691571432731 ], [ -53.458320510023448, -24.947731353549017 ],
            [ -53.458330642082387, -24.947752754208789 ], [ -53.458343029510239, -24.947773125827094 ],
            [ -53.458357558759474, -24.94779174841015 ], [ -53.458374167435259, -24.947810427547431 ],
            [ -53.458390833030208, -24.947827030494093 ], [ -53.458408273373408, -24.947847681196976 ],
            [ -53.458433777906329, -24.947880472449295 ], [ -53.458445663391416, -24.947900491269415 ],
            [ -53.458457550707529, -24.947920600287649 ], [ -53.458486849444412, -24.947955044435975 ],
            [ -53.458495999528765, -24.947976821899413 ], [ -53.458505149616634, -24.947998599361899 ],
            [ -53.458517251262606, -24.948019516963907 ], [ -53.458529453673222, -24.948040523161655 ],
            [ -53.458548570728858, -24.94805600363901 ], [ -53.458650532891355, -24.94981006246179 ],
            [ -53.457886227620996, -24.949855558242298 ], [ -53.457419115260727, -24.949915504191157 ],
            [ -53.457358893410309, -24.950054971122601 ], [ -53.441318230987513, -24.940782116208535 ],
            [ -53.437582053292523, -24.94601356559593 ], [ -53.438249787105455, -24.946224437860444 ],
            [ -53.43902156957509, -24.946470052264111 ], [ -53.441343246139333, -24.94774450656065 ],
            [ -53.439231629605082, -24.950796716748872 ], [ -53.441485785952921, -24.952088131388333 ],
            [ -53.444324180738256, -24.953712955234451 ], [ -53.449819290194647, -24.95689477757378 ],
            [ -53.450164539153782, -24.961841692391651 ], [ -53.460777461437154, -24.961244099204514 ],
            [ -53.476763737793348, -24.960334461596148 ], [ -53.477779928592561, -24.962447563565743 ],
            [ -53.478039628071535, -24.963094706325862 ], [ -53.47843789692125, -24.963673571782309 ],
            [ -53.478794236047932, -24.964041148588766 ], [ -53.479314035999074, -24.964438702373805 ],
            [ -53.480058482794711, -24.964978549132361 ], [ -53.480586192984298, -24.965344087204752 ],
            [ -53.481619235207084, -24.964771045390052 ], [ -53.482425598898757, -24.964337811453866 ],
            [ -53.481987240337403, -24.96362747744843 ], [ -53.481765841661186, -24.963078439935472 ],
            [ -53.481415911842404, -24.962293601849694 ], [ -53.480912423932239, -24.959331860637352 ],
            [ -53.480408966589536, -24.956370114790595 ] ] ] } }
    ]

}
```


Figura 5.8 – Arquivo GeoJson do bairro Centro de Cascavel.

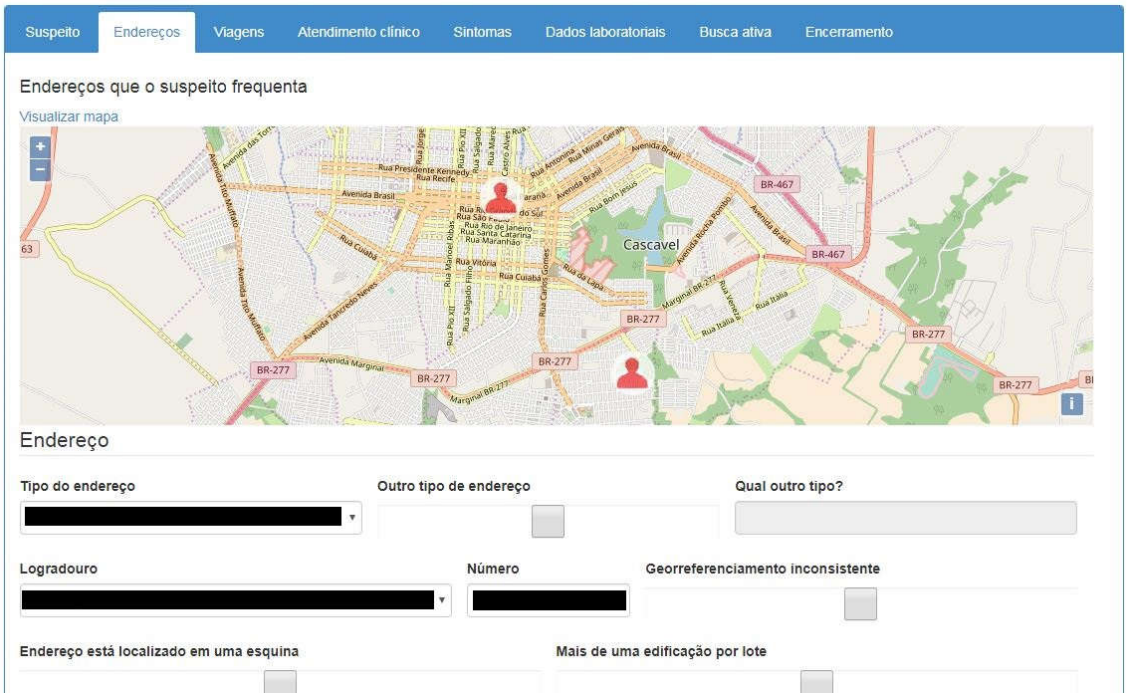


Figura 5.9 – Visualização de endereços do suspeito.

Figura 5.10 – Detalhar Ponto estratégico

Visualiza Endereço Suspeito
Controlador
Inicio
Recebe *codigo do suspeito*
Busca *endereço do suspeito*
Para cada *endereço* **faça:**
Calcula *centroide do lote*
Fim
Retorna *lista dos centroides*
Fim

View
Inicio
Recebe *lista de centroides*
Para cada *centroide* **faça**
Insere *icone* no *mapa*
Fim
Fim

Figura 5.11 – Lógica na implementação da funcionalidade Visualizar endereço do Suspeito.

Detalhar PE
View
Inicio
Recebe *coordenadas do PE* e *tipo do PE;*
Cria icone com imagem do *tipo do PE;*
Insere icone no *mapa;*
Fim

Figura 5.12 – Logica na implementação da funcionalidade detalhar PE.

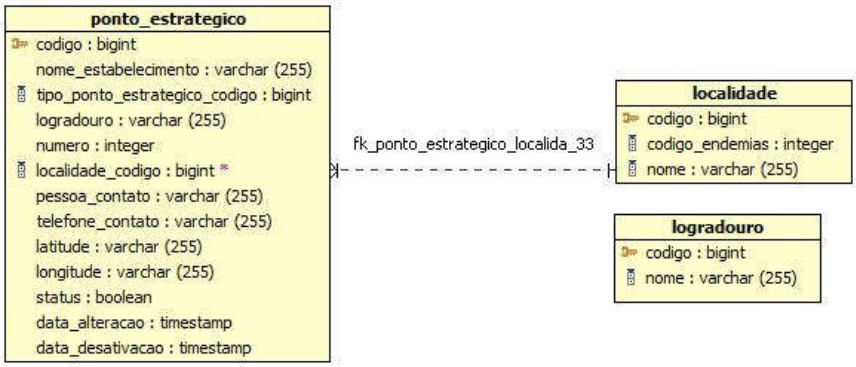


Figura 5.13 Diagrama de Entidade-Relacionamento da funcionalidade Detalhar PE.

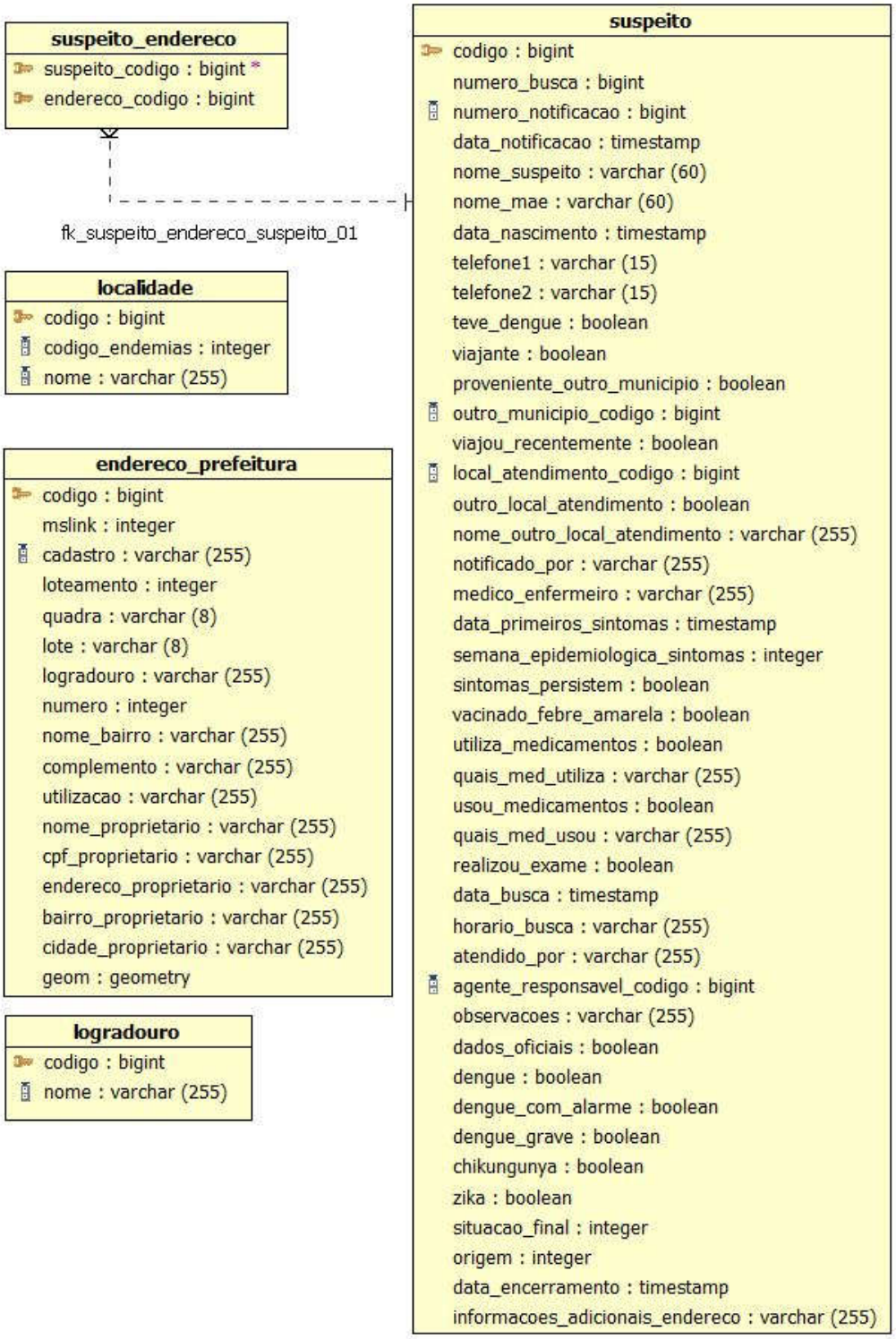


Figura 5.14 – Diagrama de Entidade-Relacionamento da funcionalidade Visualizar Endereço do Suspeito.

A apresentação do conteúdo geoespacial na camada de Visão é realizada por meio da biblioteca OpenLayers [18], tanto para as funcionalidades que se utilizam de arquitetura dual quanto as implementadas utilizando arquitetura integrada.

Para a atividade de visualização de endereços do suspeito, um *script* é disparado, assim que a página do mapa é carregada. Esse *script* captura o código do suspeito que está sendo visualizado e monta uma requisição Ajax para o servidor, como mostrado na Figura 5.15. No cabeçalho dessa requisição, dentre outras informações, é passado um Json com a informação do código do suspeito e um endereço ('/suspeitos/obtemCoordenadas'). Esse endereço é redirecionado através do arquivo de Rotas para o método responsável por implementar a busca de coordenadas na camada de Controladores, como pode ser visto na Figura 5.16.

```
var camadaSuspeitos;
var vectorIcones = [];

//Busca as coordenadas dos endereços do suspeito no banco e caso retorne com sucesso cria os icones.
function getCoordenadas() {
    var codigo = document.getElementById("codigoSuspeito").value;
    $.ajax({
        type: 'POST',
        contentType: 'application/json',
        dataType: 'json',
        data: JSON.stringify({"codigoSuspeito": codigo }),
        contentType: "application/json; charset=utf-8",
        url: '/suspeitos/obtemCoordenadas',
        success: function (json) {
            var coordenada = json;
            for (var i = 0; i < coordenada.length; i++) {
                var latitude = coordenada[i].latitude;
                var longitude = coordenada[i].longitude;

                var suspeito = new ol.Feature({
                    geometry: new ol.geom.Point(ol.proj.transform([longitude, latitude], 'EPSG:4326', 'EPSG:3857')),
                    box: false,
                    Tipo: 'Suspeito'
                });
                suspeito.setStyle(susStyle);
                vectorIcones.push(suspeito);
            }
            var vectorSuspeitos = new ol.source.Vector({
                features: vectorIcones
            });
            camadaSuspeitos = new ol.layer.Vector({
                source: vectorSuspeitos,
                style: susStyle
            });
            map.addLayer(camadaSuspeitos);
        },
        error: function (jqXHR, textStatus, errorThrown) {
            window.alert("Erro: não foi possível obter informações de endereço do suspeito");
        }
    });
}
```

Figura 5.15 – Requisição Ajax realizada para encontrar as coordenadas do endereço do suspeito

```
#SUSPEITOS ----------
#SUSPEITOS
GET     /suspeitos/                              controllers.suspeitos.SuspeitoController.lista()
POST    /suspeitos/obtemCoordenadas              controllers.suspeitos.SuspeitoController.coordenadasSuspeito()
POST    /suspeitos/obtemCoordenadasEnderecoSuspeito controllers.suspeitos.SuspeitoController.coordenadasEndereco()
GET     /suspeitos/info/:id                      controllers.suspeitos.SuspeitoController.info(id: Long)
```

Figura 5.16 – Trecho do arquivo Routes redirecionando a requisição para o método responsável.

Utilizando o código do suspeito que é recebido através do Json pela requisição Ajax, o método coordenadaSuspeito monta um formulário com os dados do suspeito requisitado, e por meio de um laço, busca por todos os endereços associados a este suspeito. Para cada par Logradouro/número encontrado é feita uma busca pelo atributo *geom* relacionado ao par na tabela *endereço_prefeitura* no banco de dados. Como pode ser visto na Figura 5.17, atributo *geom* são as propriedades geográficas do polígono que representa cada lote. Ao atributo *geom* retornado, é aplicada a função *ST_Centroid* do PostGis, que calcula o centroide de um polígono, e, portanto, seu retorno é um ponto que corresponde ao centroide do lote do suspeito.

Todos esses centroides são adicionados a uma lista de pares latitude/longitude, que é convertida em um Json e retornada pelo método. O método coordenadasSuspeito é apresentado na Figura 5.18.

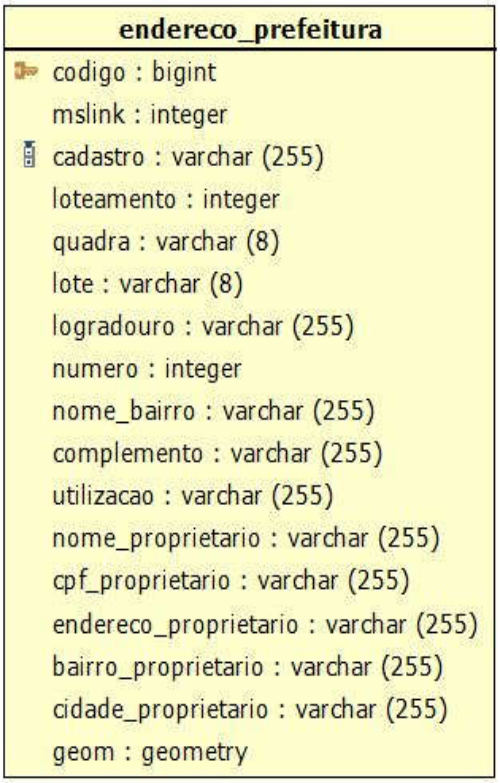
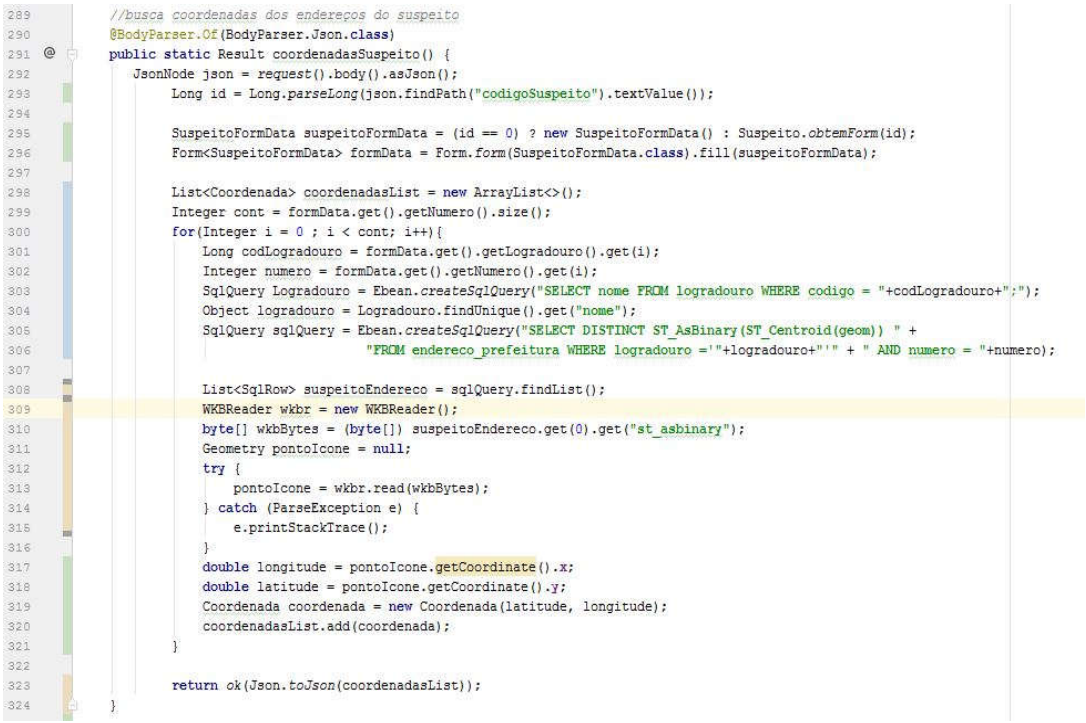


Figura 5.17 – Definição da tabela *endereço_prefeitura*.



```
289     //busca coordenadas dos endereços do suspeito
290     @BodyParser.Of(BodyParser.Json.class)
291 @   public static Result coordenadasSuspeito() {
292         JsonNode json = request().body().asJson();
293             Long id = Long.parseLong(json.findPath("codigoSuspeito").textValue());
294
295             SuspeitoFormData suspeitoFormData = (id == 0) ? new SuspeitoFormData() : Suspeito.obtemForm(id);
296             Form<SuspeitoFormData> formData = Form.form(SuspeitoFormData.class).fill(suspeitoFormData);
297
298             List<Coordenada> coordenadasList = new ArrayList<>();
299             Integer cont = formData.get().getNumero().size();
300             for(Integer i = 0 ; i < cont; i++){
301                 Long codLogradouro = formData.get().getLogradouro().get(i);
302                 Integer numero = formData.get().getNumero().get(i);
303                 SqlQuery Logradouro = Ebean.createSqlQuery("SELECT nome FROM logradouro WHERE codigo = "+codLogradouro+";");
304                 Object logradouro = Logradouro.findUnique().get("nome");
305                 SqlQuery sqlQuery = Ebean.createSqlQuery("SELECT DISTINCT ST_AsBinary(ST_Centroid(geom)) " +
306                                     "FROM endereco_prefeitura WHERE logradouro ='"+logradouro+"'" + " AND numero = "+numero);
307
308                 List<SqlRow> suspeitoEndereco = sqlQuery.findList();
309                 WKBReader wkbr = new WKBReader();
310                 byte[] wkbBytes = (byte[]) suspeitoEndereco.get(0).get("st_asbinary");
311                 Geometry pontoIcone = null;
312                 try {
313                     pontoIcone = wkbr.read(wkbBytes);
314                 } catch (ParseException e) {
315                     e.printStackTrace();
316                 }
317                 double longitude = pontoIcone.getCoordinate().x;
318                 double latitude = pontoIcone.getCoordinate().y;
319                 Coordenada coordenada = new Coordenada(latitude, longitude);
320                 coordenadasList.add(coordenada);
321             }
322
323             return ok(Json.toJson(coordenadasList));
324     }
```

Figura 5.18 – Método que retorna as coordenadas dos endereços de um suspeito.

Caso o Json retornado para a requisição Ajax pelo método coordenadasSuspeito seja uma lista válida de coordenadas, o bloco *success* mostrado na linha 36 da figura 5.15 é executado, caso contrario, o bloco *error* será executado alertando o usuário de que não foi possível obter informações de endereço do suspeito.

No bloco *success*, o conteúdo do json é atribuído a um vetor *coordenada* no javascript, e, para cada posição do vetor *coordenada* é instanciado um objeto *suspeito* do tipo *Feature* da biblioteca OpenLayers. Esta *Feature* recebe em seu atributo *Geometry* o tipo ponto (*ol.geom.Point*), e este ponto recebe como coordenadas o par latitude-longitude contidos na posição atual do vetor coordenada.

Cada um desses objetos é inserido em um vetor chamado *vectorIcones*, que será usado para criação da camada de ícones *camadaSuspeitos*. Esta camada de ícones será adicionada na lista de camadas do mapa instanciado para esta página. A instanciação do mapa para visualização dos endereços do suspeito está contida no mesmo *script* da requisição Ajax, e pode ser vista na Figura 5.19, juntamente com a instanciação da camada base, utilizada para mostrar os nomes das ruas e que utiliza como fonte a base de dados da Open Street Maps [48].

```
//Instanciação do mapa
var map = new ol.Map({
    layers: [base],
    target: 'mapa',
    view: new ol.View({
        center: center,
        zoom: 12
    })
});
```

Figura 5.19 – Instanciação do Mapa para visualização dos endereços dos suspeitos.

Para as funcionalidades de Geração do Raio e do Bloqueio a comunicação entre as camadas do MVC é feita por meio de requisições Ajax, da mesma forma que nas funcionalidades de visualização de endereços do suspeito e de pontos estratégicos, porém, os dados buscados são polígonos e não pontos. Na Figura 5.20 é apresentada a lógica na implementação da funcionalidade Novo Raio, em seguida na Figura 5.21 pode-se visualizar o trecho inicial da requisição Ajax responsável por solicitar a geração de um Raio.

Novo Raio
Controlador
Inicio
    Recebe o codigo do endereço do suspeito
    Cria buffer de 150 metros de raio com centro no *endereço do suspeito*
    Cria buffer de 170 metros de raio com centro no endereço do *suspeito*
    Para cada lote dentro do *buffer de 150 metros* **faça**:
        Adicione o *codigo do lote* a lista de *lotes pertencentes ao raio*;
    Fim
    Para cada lote dentro do buffer de 170 metros **faça**:
        Adicione o *codigo do lote* a lista de *lotes pertencentes a base*;
    Fim
    Retorna listas de lotes
Fim
view
Inicio
    Recebe *lista de lotes*
    Adiciona *lotes da base ao mapa*
    Adiciona *lotes do raio ao mapa*
Fim

Pseudocódigo 5.20 – Lógica na implementação da funcionalidade Novo Raio.

```
$.ajax({
    type: 'POST',
    contentType: 'application/json',
    dataType: 'json',
    async: false,
    data: JSON.stringify({"logradouro": logradouro, "numero": numero}),
    contentType: "application/json; charset=utf-8",
    url: '/funcoes/retornaRaio',
    success: function (json){
        var polisRaio = [];
        var polisBase = [];

        var vectorSourceRaio = new ol.source.Vector();
        var vectorSourceBase = new ol.source.Vector();
        var raio = JSON.parse(json);

        var lotesRaio = raio[0];          //posição 0 do Array Raio contém as geometrias dos lotes pertencentes ao raio
        var mslink = raio[1];             //posição 1 do Array Raio contém os mslinks dos lotes pertencentes ao raio
        var numero = raio[2];             //posição 2 do Array Raio contém os numeros dos lotes pertencentes ao raio
        var logradouro = raio[3];         //posição 3 do Array Raio contém os logradouros dos lotes pertencentes ao raio
        var lotesBase = raio[4];          //posição 4 do Array Raio contém as geometrias dos lotes pertencentes a base
        var mslinkBase = raio[5];         //posição 5 do Array Raio contém os mslinks dos lotes pertencentes a base
        var numeroBase = raio[6];         //posição 6 do Array Raio contém os numeros dos lotes pertencentes a base
        var logradouroBase = raio[7];     //posição 7 do Array Raio contém os logradouros dos lotes pertencentes a base
```

Figura 5.21 – Cabeçalho e inicio do bloco success da requisição Ajax para geração de Raio

No trecho apresentado na Figura 5.21, é possível visualizar na linha 66, o cabeçalho da requisição Ajax montando um Json com os atributos logradouro e número que serão passados para o método *retornaRaio* da camada Controlador. O inicio do método *retornaRaio* pode ser visto na Figura 5.22, quebras de linha foram inseridas para melhor visualização do código da imagem.

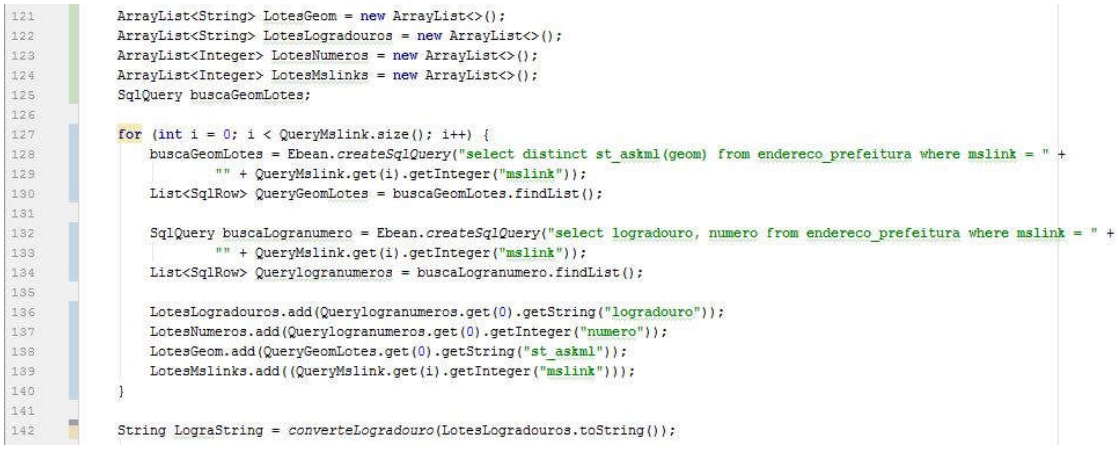
```
53      @BodyParser.Of(BodyParser.Json.class)
54      public static Result retornaRaio() {
55          JsonNode json = request().body().asJson();
56          try {
57              Long codigoLogradouro = Long.parseLong(json.findPath("logradouro").textValue());
58              Integer numero = Integer.parseInt(json.findPath("numero").textValue());
59
60              try {
61                      Logradouro logradouro = Logradouro.find.byId(codigoLogradouro);
62                      List<EnderecoPrefeitura> enderecosPrefeitura = EnderecoPrefeitura.find.where().eq("logradouro" +
63                              "", logradouro.getNome()).eq("numero", numero).findList();
64                      if (enderecosPrefeitura.size() == 0 || enderecosPrefeitura.get(0) == null)
65                          return badRequest("Não foi possivel obter informações");
66
67                      //Obtém codigo do lote
68                      SqlQuery buscaCodigo = Ebean.createSqlQuery("select codigo from endereco_prefeitura where logradouro='" +
69                              "" + logradouro + "' and numero = " + numero);
70                      List<SqlRow> QueryCodigo = buscaCodigo.findList();
71
72                      SqlQuery buscaMslinks = Ebean.createSqlQuery("select distinct b.mslink from" +
73                              " endereco_prefeitura a, endereco_prefeitura b where a.codigo =" + QueryCodigo.get(0).getInteger("codigo") + "" +
74                              " and st_intersects(st_transform(st_buffer(st_transform(a.geom, 29192), 150.0), 4326), b.geom)");
75                      List<SqlRow>QueryMslink = buscaMslinks.findList();
76
77                      SqlQuery buscaMslinksBase = Ebean.createSqlQuery("select distinct b.mslink from " +
78                              "endereco_prefeitura a, endereco_prefeitura b where a.codigo ="+QueryCodigo.get(0).getInteger("codigo")+"" +
79                              " and st_intersects(st_transform(st_buffer(st_transform(a.geom, 29192), 170.0), 4326), b.geom)");
80                      List<SqlRow>QueryMslinkBase = buscaMslinksBase.findList();
```

Figura 5.22 – Trecho inicial do método *retornaRaio.*

No trecho apresentado na Figura 5.22, as informações de código do logradouro e número enviados pela requisição Ajax são utilizados para definir o centro do Raio, caso este endereço seja um endereço válido. Em seguida, na linha 72, uma busca é realizada utilizando as funções do PostGis calculando a intersecção entre um *buffer* circular de 150 metros de raio com centro no lote do suspeito e todos os outros lotes presentes na tabela *endereço_prefeitura.* O retorno dessa consulta é uma lista com os *mslinks* de todos os lotes pertencentes a essa intersecção.

Em seguida uma segunda consulta é realizada utilizando um buffer de 170 metros de raio, os lotes retornados por essa segunda consulta serão utilizados como sugestão para que o usuário decida se serão adicionados à atividade ou não. Essa camada de lotes será chamada de raio base no decorrer deste texto.

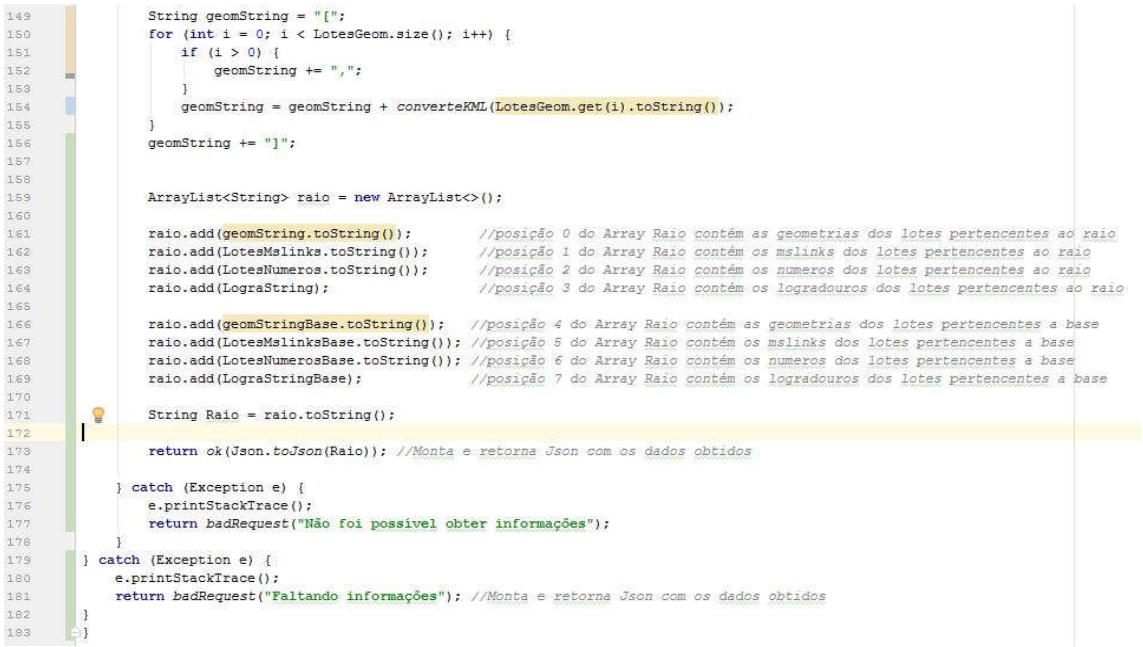
As listas com os *mslinks* pertencentes ao raio e ao raio base serão iteradas realizando consultas ao banco de dados na tabela *endereço_prefeitura* buscando os atributos geometria, logradouro e número de cada lote, como pode ser visto na Figura 5.23.



```
ArrayList<String> LotesGeom = new ArrayList<>();
ArrayList<String> LotesLogradouros = new ArrayList<>();
ArrayList<Integer> LotesNumeros = new ArrayList<>();
ArrayList<Integer> LotesMslinks = new ArrayList<>();
SqlQuery buscaGeomLotes;

for (int i = 0; i < QueryMslink.size(); i++) {
    buscaGeomLotes = Ebean.createSqlQuery("select distinct st_askml(geom) from endereco_prefeitura where mslink = " +
            "" + QueryMslink.get(i).getInteger("mslink"));
    List<SqlRow> QueryGeomLotes = buscaGeomLotes.findList();

    SqlQuery buscaLogranumero = Ebean.createSqlQuery("select logradouro, numero from endereco_prefeitura where mslink = " +
            "" + QueryMslink.get(i).getInteger("mslink"));
    List<SqlRow> Querylogranumeros = buscaLogranumero.findList();

    LotesLogradouros.add(Querylogranumeros.get(0).getString("logradouro"));
    LotesNumeros.add(Querylogranumeros.get(0).getInteger("numero"));
    LotesGeom.add(QueryGeomLotes.get(0).getString("st_askml"));
    LotesMslinks.add((QueryMslink.get(i).getInteger("mslink")));
}

String LograString = converteLogradouro(LotesLogradouros.toString());
```

Figura 5.23 – Trecho do método *retornaRaio* realizando a busca pelas informações dos lotes pertencentes ao raio.

Os atributos de cada lote são adicionados às listas específicas e convertidos para *strings* para que possa ser montado um Json com o resultado, como mostrado na figura 5.24. O atributo geometria precisou ser tratado através do método *converteKML*, para ficar em um formato parecido com os atributos de um arquivo geoJson, a implementação do método *converteKML* é apresentado na Figura 5.25.



```
    String geomString = "[";
    for (int i = 0; i < LotesGeom.size(); i++) {
        if (i > 0) {
            geomString += ",";
        }
        geomString = geomString + converteKML(LotesGeom.get(i).toString());
    }
    geomString += "]";


    ArrayList<String> raio = new ArrayList<>();

    raio.add(geomString.toString());        //posição 0 do Array Raio contém as geometrias dos lotes pertencentes ao raio
    raio.add(LotesMslinks.toString());      //posição 1 do Array Raio contém os mslinks dos lotes pertencentes ao raio
    raio.add(LotesNumeros.toString());      //posição 2 do Array Raio contém os numeros dos lotes pertencentes ao raio
    raio.add(LograString);                  //posição 3 do Array Raio contém os logradouros dos lotes pertencentes ao raio

    raio.add(geomStringBase.toString());   //posição 4 do Array Raio contém as geometrias dos lotes pertencentes a base
    raio.add(LotesMslinksBase.toString()); //posição 5 do Array Raio contém os mslinks dos lotes pertencentes a base
    raio.add(LotesNumerosBase.toString()); //posição 6 do Array Raio contém os numeros dos lotes pertencentes a base
    raio.add(LograStringBase);             //posição 7 do Array Raio contém os logradouros dos lotes pertencentes a base

    String Raio = raio.toString();

    return ok(Json.toJson(Raio)); //Monta e retorna Json com os dados obtidos

} catch (Exception e) {
    e.printStackTrace();
    return badRequest("Não foi possível obter informações");
}
} catch (Exception e) {
    e.printStackTrace();
    return badRequest("Faltando informações"); //Monta e retorna Json com os dados obtidos
}
}
```

Figura 5.24 – Trecho final do método *retornaRaio*.

```
175     public static String converteKML(String string) {
176         string = string.replace("<MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>", "[{\"lng\":");
177         string = string.replace(",",",\"lat\":");
178         string = string.replace(" ","},{\"lng\":");
179         string = string.replace("</coordinates></LinearRing></outerBoundaryIs></Polygon></MultiGeometry>","}]");
180         return string;
181     }
```

Figura 5.25 – Implementação do método *converteKML*

O Json retornado pelo método *retornaRaio* é recebido como resposta da requisição Ajax apresentada na Figura 5.21, e se for um Json válido, o bloco *success* é executado, atribuindo cada posição do Json para um vetor javascript, para facilitar a leitura do código subsequente. A Figura 5.26 apresenta o trecho do bloco *success* onde as geometrias dos lotes do raio e do raio base são definidas utilizando o conjunto de coordenadas dos vértices retornados pelo atributo geometria.

```
86          // define geometria dos lotes da base
87          for(var i = 0; i < lotesBase.length; i++){
88              var poli = lotesBase[i];
89              polisBase.push(poli.map(function(item){
90                  var transform = ol.proj.transform([item.lng, item.lat], 'EPSG:4326', 'EPSG:3857');
91                  return[transform[0], transform[1]];
92              }))
93          }
94          //define geometria dos lotes pertencentes ao raio
95          for(var i = 0; i < lotesRaio.length; i++){
96              var poli = lotesRaio[i];
97              polisRaio.push(poli.map(function(item){
98                  var transform = ol.proj.transform([item.lng, item.lat], 'EPSG:4326', 'EPSG:3857');
99                  center = transform;
100                 return[transform[0], transform[1]];
101             }))
102         }
```

Figura 5.26 – Definição das geometrias dos lotes pertencentes ao raio e raio base.

Após a definição das geometrias, é realizada a instanciação de um objeto *Polygon* da biblioteca OpenLayers para cada geometria definida no passo anterior, tanto para o raio quanto para o raio base, porém no laço de instanciação dos *Polygons* do raio, também é gerada uma tabela HTML onde serão exibidos os atributos dos lotes pertencentes ao raio e a opção de remoção de lotes individualmente. Isto é possível pois, durante a instanciação dos *Polygons,* cada polígono recebe como identificador o número do *mslink* do lote que ele representa, também são adicionados os atributos número e logradouro. Para cada execução do laço de instanciação, uma linha é adicionada na tabela com os atributos do lote instanciado e um botão é gerado com uma chamada para a função *removeLote* e passando o *mslink* do lote atual como parâmetro. Todo este processo é apresentado na Figura 5.27.

```
103     //adiciona os lotes a base ao vectorSource do mapa
104     for(var i = 0; i < polisBase.length; i++) {
105         var polygon = new ol.geom.Polygon([polisBase[i]]);
106 
107         var feature = new ol.Feature(polygon);
108         feature.setId(mslinkBase[i]);
109         feature.set("numero",numeroBase[i]);
110         feature.set("logradouro", logradouroBase[i]);
111         vectorSourceBase.addFeature(feature);
112     }
113 
114     //adiciona os lotes do raio ao vectorSource do mapa, e cria a tabela de endereços
115     for(var i = 0; i < polisRaio.length; i++) {
116         var polygon = new ol.geom.Polygon([polisRaio[i]]);
117 
118         var feature = new ol.Feature(polygon);
119         feature.setId(mslink[i]);
120         feature.set("numero",numero[i]);
121         feature.set("logradouro", logradouro[i]);
122         vectorSourceRaio.addFeature(feature);
123 
124         var table = document.getElementById('tabelaLotes');
125         var row = table.insertRow(i+1);
126         row.innerHTML = "<td>"+mslink[i]+"</td><td>"+logradouro[i]+"</td><td>"+numero[i]+"" +
127             "</td><td align='center' ><input type='button' value='x' class='remover-linha" +
128             "' onclick='removeLote("+mslink[i]+")'></td>";
129     }
```

Figura 5.27 – Instanciação dos *Polygons* e inserção na tabela de endereços.

Em seguida os vetores de Polygons são utilizados como fonte para a instanciação de uma *LayerVector.* Uma *LayerVector* é a classe que representa uma camada de objetos que serão adicionados ao mapa, e, como pode ser visto na Figura 5.28, as camadas *vectorLayerRaio* e *vectorLayerBase* são adicionadas ao mapa e cada um dos polígonos pertencentes a essas camadas recebem um conjunto de propriedades como cor de preenchimento, cor de contorno, estilo da fonte, cor da fonte e o mslink de cada polígono é definido como texto de legenda para os polígonos da camada raio, os polígonos da camada raio base recebem um conjunto de estilos diferentes. O resultado da criação do raio é apresentado na Figura 5.29 e o diagrama de Entidade –Relacionamento das tabelas envolvidas no processo é mostrado na Figura 5.30, as mesmas entidades são utilizadas na implementação da funcionalidade Alterar Raio. Esta atividade é muito importante operacionalmente para o setor de endemias, como já explicado no capitulo 4, e a automatização por meio desta funcionalidade será fundamental para agilizar as atividades desenvolvidas.

```
//instancia os vectorlayers das camadas Base e Raio
var vectorLayerRaio = new ol.layer.Vector({ source: vectorSourceRaio});
var vectorLayerBase = new ol.layer.Vector({ source: vectorSourceBase});
//adiciona as camadas Base e Raio ao mapa
mapa.addLayer(vectorLayerBase);
mapa.addLayer(vectorLayerRaio);
//Aplica o estilo e legenda para a camada Raio, o índice 2 passado no
// parametro GetArray se refere ao Array de layers do mapa composto por 3 camadas [OSM, Base, Raio]
var featuresRaio = mapa.getLayers().getArray()[2].getSource().getFeatures();
for(var i = 0; i< featuresRaio.length; i++){
    var id = featuresRaio[i].getId();
    var estilo = new ol.style.Style({
        fill: new ol.style.Fill({
            color: '#D2691E'
        }),
        stroke: new ol.style.Stroke({
            color: '#0000ff',
            width: 1
        }),
        text: new ol.style.Text({
            text: new ol.style.Text({
                text: id,
                font: '12px Calibri,sans-serif',
                fill: new ol.style.Fill({
                    color: '#000'
                })
            })
        })
    })
    estilo.getText().setText(""+id);
    featuresRaio[i].setStyle(estilo);
};
```

Figura 5.28 – Inserção das camadas raio e raio base ao mapa e definição de estilos.

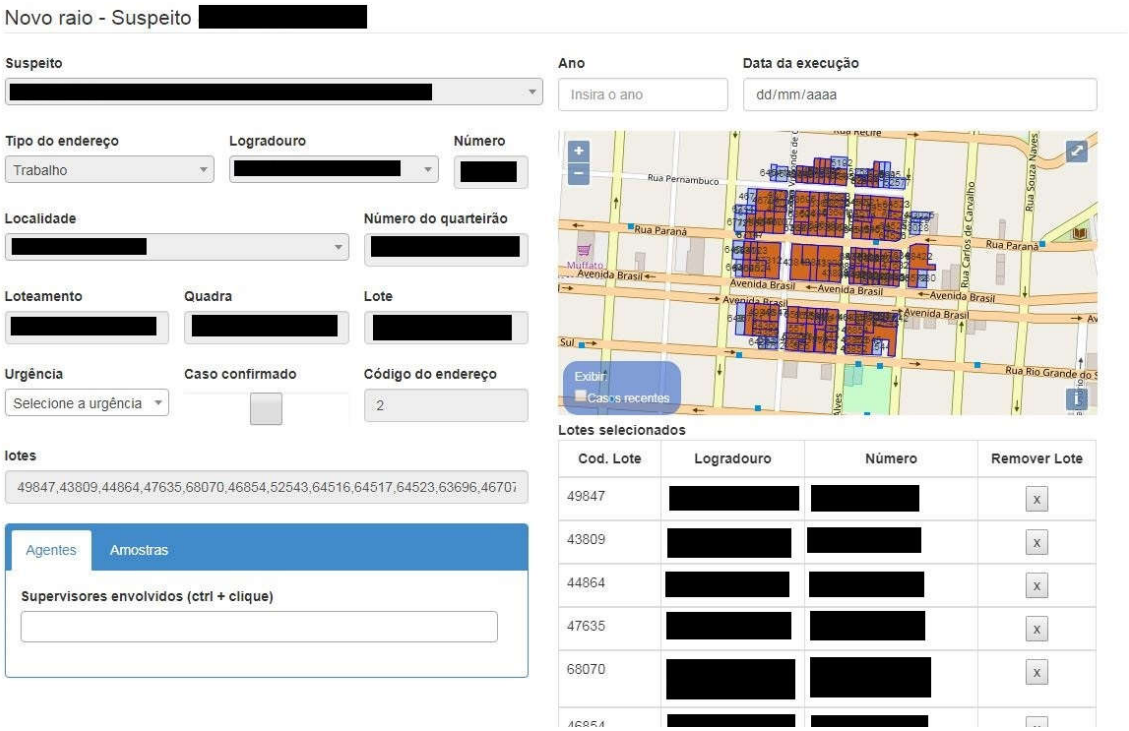


Figura 5.29 – Tela da funcionalidade Novo Raio

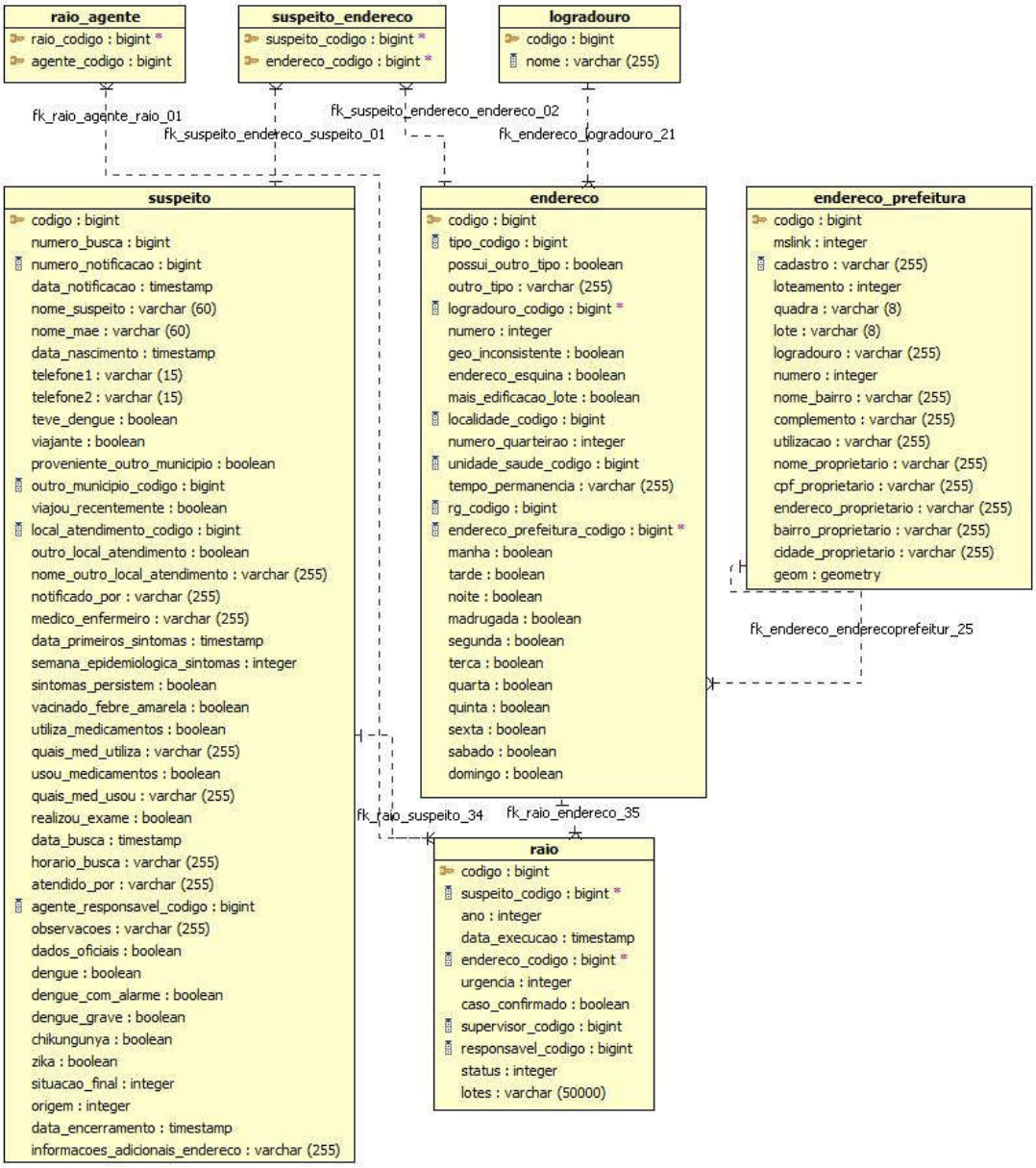


Figura 5.30 – Diagrama de Entidade- Relacionamento das funcionalidades novo Raio e Alterar Raio.

Para o usuário adicionar um novo lote a camada raio ele deve clicar no lote da camada raio base que deseja adicionar, no momento em que é detectado um clique no mapa a função *adicionaLote* é chamada e o pixel que recebeu o clique é passado como parâmetro, como pode ser visto na Figura 5.31. Dentro da função *adicionaLote* são o pixel recebido como parâmetro é utilizado para detectar se o clique aconteceu sobre um polígono da camada raio base capturar o identificador deste polígono e verificar se este polígono já faz parte da camada raio. Caso não esteja contido na camada raio é gerada uma cópia deste polígono e esta cópia recebe

o conjunto de estilos dos polígonos da camada raio, uma linha com os atributos deste polígono é adicionada à tabela de endereços e por fim a função *atualizaLotes* é chamada para adicionar o mslink deste polígono à lista de *mslinks* dos lotes pertencentes ao raio que é exibida no campo lotes na tela de criação de raio. A implementação da função *atualizaLotes* é apresentada na imagem 5.32.

A lista presente no campo lotes é o que será gravado no banco de dados quando a edição do raio for concluída e salva.

```
mapa.on('click', function (evt) {
    adicionaLote(evt.pixel);
});

var adicionaLote = function (pixel) {
    var lote = mapa.forEachFeatureAtPixel(pixel, function (lote) {
        return lote;
    });
    var mslink = lote.getId();
    var source = mapa.getLayers().getArray()[2].getSource();
    var colecao = source.getFeatureById(mslink);
    if(!colecao) {
        var novoLote = lote.clone();
        var logradouro = lote.get('logradouro');
        var numero = lote.get('numero');
        novoLote.set("logradouro", logradouro);
        novoLote.set("numero", numero);
        novoLote.setId(mslink);
        var estilo = new ol.style.Style({
            fill: new ol.style.Fill({
                color: '#7FFF00'
            }),
            stroke: new ol.style.Stroke({
                color: '#0000ff',
                width: 1
            }),
            text: new ol.style.Text({
                text: new ol.style.Text({
                    font: '12px Calibri,sans-serif',
                    fill: new ol.style.Fill({
                        color: '#000'
                    })
                })
            })
        });
        estilo.getText().setText("" + mslink);
        novoLote.setStyle(estilo);
        source.addFeature(novoLote);
        //adiciona a linha da tabela de lotes com as informações do lote adicionado
        var table = document.getElementById('tabelaLotes');
        var row = table.insertRow(1);
        row.innerHTML = "<td id='mslink'>" + mslink + "</td><td>" + logradouro + "</td><td>" + numero +
            "</td><td align='center'><input type='button' value='x' class='remover-linha' onclick='removeLote("+mslink+")'></td>";
        //atualiza os lotes selecionados para o raio
        atualizaLotes()
    }
};
```

Figura 5.31 – Implementação da funcionalidade *adicionaLote.*

A função removeLotes é chamada pelos botões criados dentro da tabela de endereços, cada botão está associado a um endereço e quando clicado envia como parâmetro para a função *removeLotes* o *mslink* do lote que será removido do raio. O polígono presente na camada raio que possui o mslink passado como parâmetro é selecionado e removido, em seguida a linha da tabela de endereços correspondente ao endereço é removida e a função *atualizaLotes* é chamada para atualizar a lista de lotes pertencentes ao raio após a exclusão. A implementação da remoção de lotes pode ser vista na Figura 5.32.

```
208  //função chamada pelo botão remover lote da tabela
209  function removeLote(id) {
210      //remove a feature correspondente ao lote no mapa
211      var layers = mapa.getLayers();
212      var source = layers.getArray()[2].getSource();
213      var feature = source.getFeatureById(id);
214      source.removeFeature(feature);
215      //remove a linha correspondente ao lote na tabela
216      $('#tabelaLotes').on('click', '.remover-linha', function () {
217          $(this).closest('tr').remove();
218          console.log('removeu');
219          atualizaLotes()
220      });
221  };
```

Figura 5.32 – Implementação da funcionalidade remover lote do raio.

Para a funcionalidade de edição de Raio o processo é simplificado, pois a lista de mslinks dos lotes pertencentes ao raio é salva no banco de dados, o que elimina a necessidade de recalcular os lotes pertencentes. A Figura 5.33 apresenta a tela da funcionalidade de edição de Raio e a Figura 5.34 apresenta a lógica na implementação da funcionalidade.

Para a funcionalidade de bloqueio o mesmo principio da funcionalidade edição de raio é aplicado, como pode ser visto no pseudocódigo da Figura 5.35, visto que a atividade de bloqueio sempre será realizada na mesma região em que o raio que a gerou foi executado. Então utilizando das informações de mslinks registradas no banco a partir da criação do raio, a visualização da região do bloqueio é construída. A Figura 5.36 apresenta a tela de edição de bloqueio e o diagrama de Entidade-Relacionamento da funcionalidade é apresentado na Figura 5.37.

Alterar raio
Controlador
Inicio
    Busca *lotes pertencentes ao raio*
    Cria *buffer de 170 metros de raio com centro no endereço do suspeito*
    Para cada *lote dentro do buffer base* **faça:**
        Adiciona *lote* na *lista de lotes da base*
    Fim
    Retorna *as listas de lotes*
Fim

View
Inicio
    Recebe *listas de lotes*
    Adiciona *lotes da base* ao *mapa*
    Adiciona *lotes do raio* ao *mapa*
Fim

Figura 5.33 – Lógica na implementação da funcionalidade Alterar Raio.

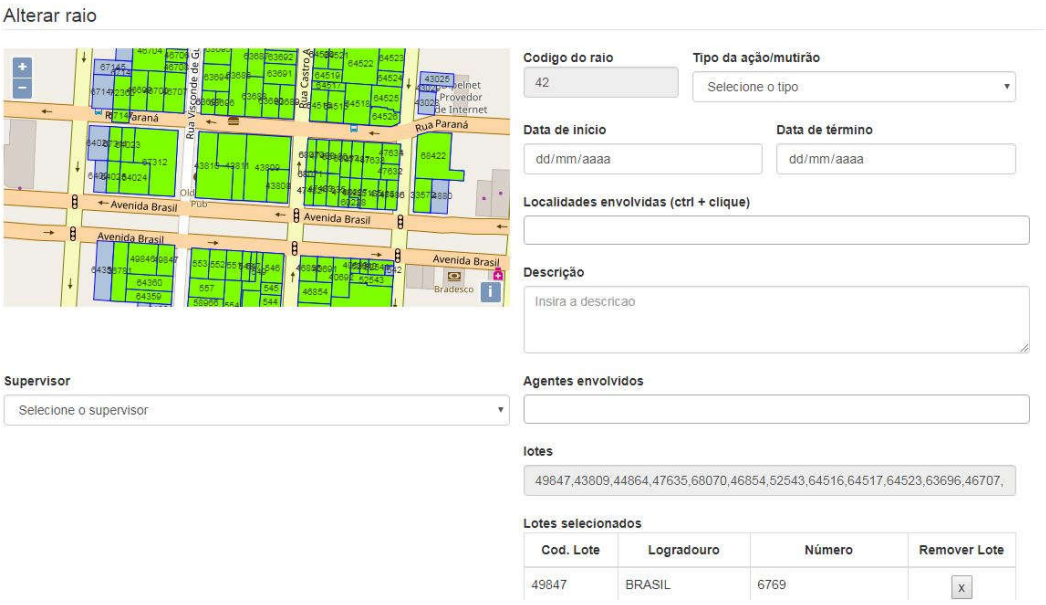


Figura 5.34 – Tela da funcionalidade Alterar Raio.

Alterar Bloqueio
Controlador
Inicio
Busca *lotes pertencentes ao bloqueio*
Cria *buffer de 170 metros de raio com centro no endereço do suspeito*
Para cada *lote dentro do buffer base* **faça:**
Adiciona *lote* na *lista de lotes da base*
Fim
Retorna *as listas de lotes*
Fim

View
Inicio
Recebe *listas de lotes*
Adiciona *lotes da base* ao *mapa*
Adiciona *lotes do bloqueio* ao *mapa*
Fim

Figura 5.35 – Lógica na implementação da funcionalidade alterar Bloqueio.

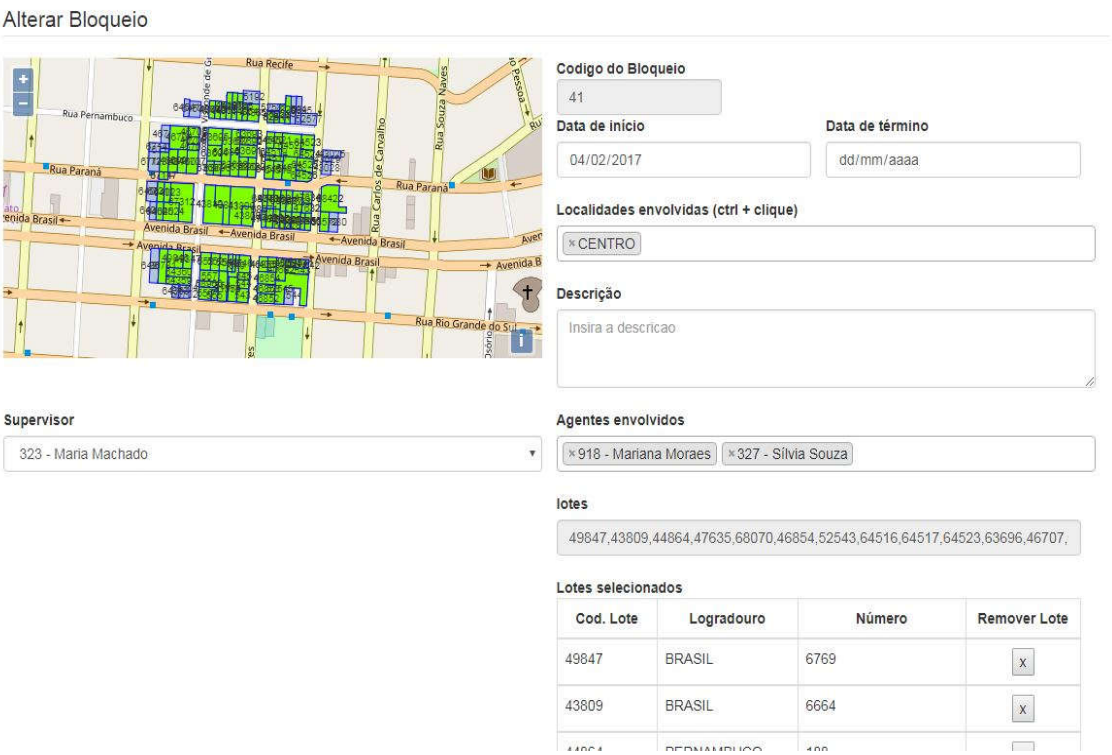


Figura 5.36 – Tela da funcionalidade Alterar Bloqueio.

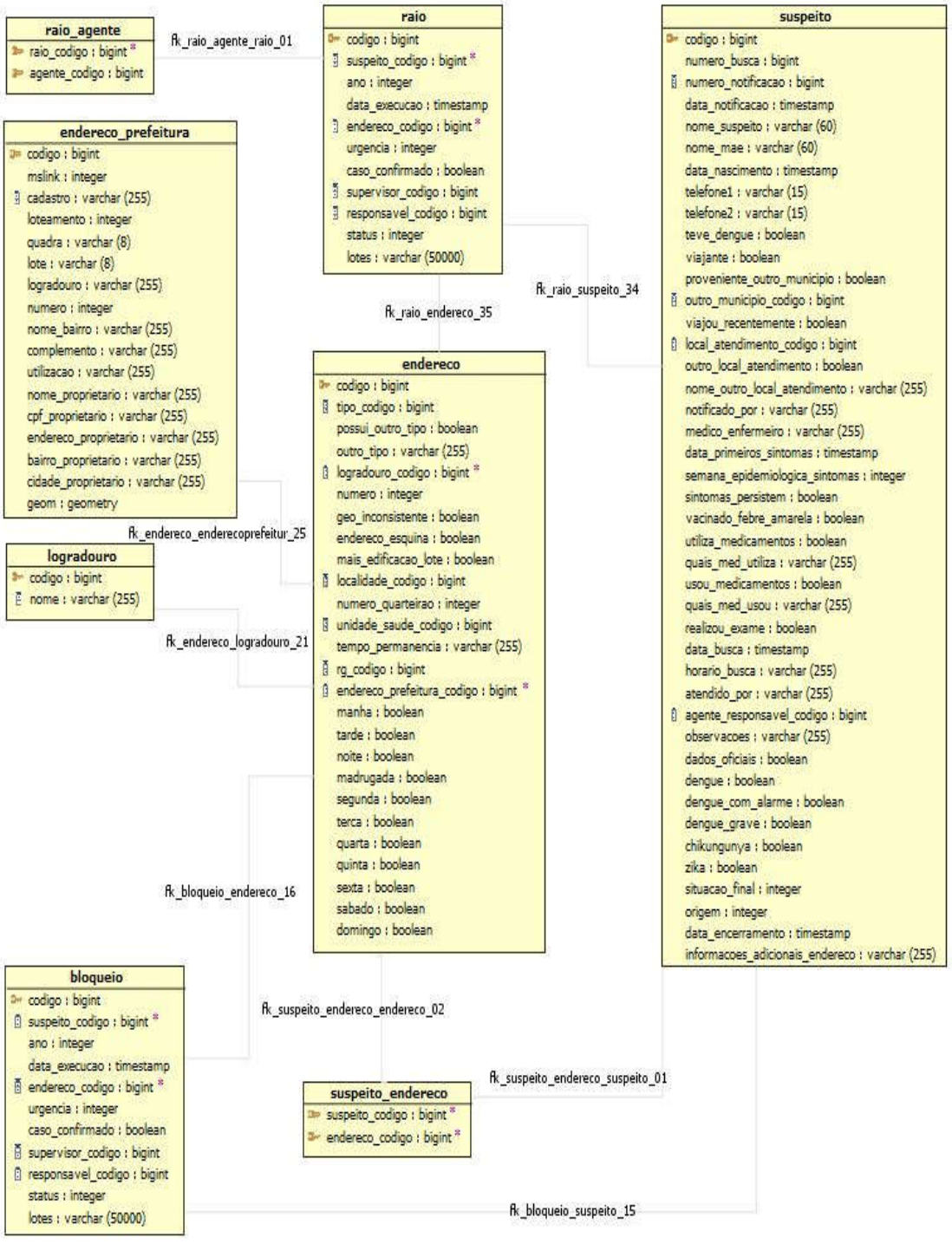


Figura 5.37 – Diagrama de entidade-relacionamento da funcionalidade Alterar Bloqueio.

Capítulo 6

Considerações Finais

Neste capítulo são discutidos os principais aspectos de seu desenvolvimento, incluindo dificuldades encontradas, verificação do resultado final das funcionalidades implementadas e a necessidade de trabalhos futuros.

6.1 Execução dos Objetivos

Considerando os objetivos gerais e específicos elencados no capítulo 1, a seguir são detalhadas as formas como a execução dos objetivos específicos se deram.

a) Realizar revisão bibliográfica sobre sistemas de informações geográficos que empreguem tecnologias web, mas necessariamente, OpenLayers, PostgreSQL, PostGIS:

Para a execução desta etapa do trabalho foi realizada uma pesquisa por artigos, trabalhos e sistemas de informações geográficos voltados para gerenciamento de dados sobre Dengue, Chikungunya e Zika e que utilizassem as tecnologias escolhidas. Os sistemas encontrados estão listados no capítulo 3 deste trabalho.

A principal dificuldade nesta etapa foi a não especificação das tecnologias utilizadas no desenvolvimento da maior parte dos sistemas encontrados e a indisponibilidade para testar esses sistemas, o que limita a possibilidade de uma comparação técnica entre o SIGAEDES e esses sistemas.

b) Identificar e especificar as funcionalidades que requeiram geovisualização no SIGAEDES:

Esta etapa foi executada utilizando a metodologia de prototipação. Durante o processo de criação do protótipo não funcional de todo o SIGAEDES as funcionalidades com necessidades de geovisualização foram sendo identificadas e priorizadas, o que facilitou a etapa de especificação e também de verificação dos requisitos junto aos agentes do Setor de Controle de Endemias.

c) Implementar e testar as funcionalidades que requeiram geovisualização no SIGAEDES:

A etapa de implementação e teste das funcionalidades foi a que demandou a maior parcela de tempo do cronograma, principalmente pela necessidade de aprendizado das tecnologias e ferramentas, mas também pelas necessidades de alterações nas funcionalidades

que não haviam sido previstas na etapa de prototipação. Um exemplo de alteração é a mudança no processo de cadastro de Reconhecimento Geral, que no protótipo foi previsto utilizando um mapa de lotes de toda a cidade, porém somente depois da funcionalidade ser implementada percebeu-se a necessidade de alteração, como é detalhado no capítulo 5. Mesmo que essas alterações tenham requerido tempo de cronograma com retrabalho, foi possível implementar e testar todas as funcionalidades previstas.

d) Avaliar e analisar os resultados:

Durante a etapa de planejamento foi definido que a avaliação e análise dos resultados seriam realizadas por meio de questionários respondidos pelos *stakeholders* que testaram o sistema. No entanto, para que isso aconteça o sistema deveria ser disponibilizado para uso dos agentes e supervisores do Setor de Controle de Endemias, o que não aconteceu devido a diversas dificuldades quanto à efetivação. Algumas reuniões foram agendadas para a verificação das funcionalidades já implementadas e implantação do sistema para testes, porém devido à agenda dos gestores essas reuniões acabaram por ser adiadas algumas vezes, o que acabou extrapolando limite do cronograma.

Como tentativa de minimizar a falta de análise dos resultados, uma apresentação do sistema em funcionamento foi realizada no dia 21 de novembro de 2017 para alguns supervisores do Setor de Controle de Endemias, nesta reunião estavam presentes a coordenadora geral da Vigilância Ambiental, sra. Clair Wagner, a coordenadora do Controle de Endemias sra. Ana Paula Barbosa Iyrkievicz e alguns agentes Supervisores de Campo, todos vinculados à Prefeitura Municipal de Cascavel.

O foco principal desta apresentação foram as funcionalidades com visualizações geoespaciais e seu funcionamento. A opinião dos participantes foi positiva quanto aos resultados obtidos. Segundo eles, o SIGAEDES superou as expectativas principalmente pelos fatores de velocidade e qualidade das informações disponibilizadas pelo sistema. Os agentes demonstraram motivação em continuar contribuindo para o desenvolvimento e implantação do sistema, mas também manifestaram preocupação com relação à demanda do Setor quanto à equipamentos, maior velocidade de acesso à internet, e responsabilidade quanto à manutenção do sistema.

Através da execução de cada um desses objetivos específicos, pode-se concluir que o objetivo geral do trabalho também foi alcançado, que foi desenvolver soluções de consultas e visualizações funcionais, eficientes para o SIGAEDES e úteis para o Controle de Endemias, utilizando tecnologias livres.

6.2 Trabalhos Futuros

Como trabalhos futuros para o SIGAEDES foram identificadas, dentre outras, as seguintes atividades:

a) implantação definitiva do sistema para uso dos gestores e agentes do Setor de Controle de Endemias, o que depende de melhorar a velocidade de banda de internet atual do Setor, disponibilização de um servidor pela prefeitura para rodar o sistema;

b) implementação de relatórios em mapa específicos conforme forem percebidas suas necessidades durante o uso do sistema pelos gestores;

c) continuar melhorando o sistema e integrando novas funcionalidades incrementalmente em conjunto com os gestores e agentes do setor.

Apêndice A

Listagem das Tabelas do Banco de Dados do Sistema

acao_mutirao
acao_mutirao_agente
acao_mutirao_localidade
acesso
agente
agente_cargo
agente_curso
agente_equipamento
agente_filho
amostra
ano_epidemiologico
ano_epidemiologico_sem
ana_epidem
armadilha
atualizacao_sinan
atualizacao_sinan_inconsi
stencia
atualizacao_sinan_suspeit
o_sinan
atualizacao_sinan_suspeit
o_zika_
bairro
bloqueio
bloqueio_agente
campanha
cargo
curso
dia_semana
dispositivo_movel
endereco
endereco_prefeitura
equipamento
equipe
equipe_agente
escolaridade
estado
estrato
estrato_localidade
exame_laboratorial
fad_atividade
fad_visita
filho
imovel
inconsistencia_suspeito_si
nan
local_atendimento
localidade
logradouro
municipio
nivel_acesso
pais
play_evolutions
ponto_estrategico
quarteirao
raio
raio_agente
reconhecimento_geral
sangramento
semana_epidemiologica
sintoma
solicitacao_alteracao_rg
spatial_ref_sys
suspeito
suspeito_endereco
suspeito_exame_laborator
ial
suspeito_sangramento
suspeito_sinan
suspeito_sinandengue
suspeito_sintoma
suspeito_viagem
suspeito_zika_sinan
tipo_acao_mutirao
tipo_armadilha
tipo_atividade
tipo_deposito
tipo_endereco
tipo_imovel
tipo_irregularidade
tipo_ponto_estrategico
tratamento_especial
turno
unidade_saude
uniforme
usuario
viagem

Apêndice B

Formulário do Boletim de Reconhecimento Geral

MINISTÉRIO DA SAUDE
FUNDAÇÃO NACIONAL DE SAUDE
Programa de controle da febre amarela e dengue – PCFAD

Boletim de Reconhecimento

RESPONSAVEL			
Ispetor Geral		Inspetor	
Cheque de Equipe		Agente	

UF					
DISTRITO		SUBDISTR.			
MUNICIPO		SUBLOCAL			
LOCALIDADE		CATEGORIA		QUART Nº	

RUA OU LOGRADOURO	Número	Sequencial	Complemento	Lado	Tipo do Imóvel

RUA OU LOGRADOURO	Número	Sequencial	Complemento	Lado	Tipo do Imóvel

FAD 05

Fechamento

Residencial	R	
Comercio	C	
Terreno Baldio	TB	

Ponto Estratégico	PE	
Outros	O	
Total Geral		

Nome	
Assinatura	

Data	

Referências Bibliográficas

[1] Instituto Oswaldo Cruz. Dengue Virus e Vetor. Disponível em: <http://www.ioc.fiocruz.br/dengue/textos/sobreovirus.html>. Acesso em 18 ago. 2017.

[2] Dengue: guidelines for diagnosis, treatment, prevention and control - New edition. World Health Organization: 2009. Disponível em: <http://whqlibdoc.who.int/publications/2009/9789241547871_eng.pdf> Acesso em 15 dez. 2013.

[3] World Health Organization. Dengue and severe dengue. Disponível em: <http://www.who.int/mediacentre/factsheets/fs117/en/>. Acesso em 16 dez. 2013.

[4] Ministério da Saúde (MS). Preparação e resposta à introdução do vírus Chikungunya no Brasil. Brasília : Ministério da Saúde, 2014. 102 p.

[5] Ministério da Saúde (MS) - Secretaria de Vigilância em Saúde. Boletim Epidemiológico nº 47. Monitoramento dos casos de dengue, febre de chikungunya e febre pelo vírus Zika até a Semana Epidemiológica 52, 2015. http://portalsaude.saude.gov.br/images/pdf/2016/janeiro/15/svs2016-be003-dengue-se52.pdf

[6] Organização Pan-Americana da Saúde (OPAS). Zika nas Américas: confira as respostas para as perguntas mais frequentes sobre o vírus. Disponivel em: <http://www.paho.org/bra/index.php?option=com_content&view=article&id=4978:zika-nas-ama-copy-ricas-confira-as-respostas-para-as-perguntas-mais-frequentes-sobre-o-va-rus&Itemid=816> Acesso em 11 mai. 2017.

[7] World Health Organization (WHO). Zika situation report. Disponível em: <http://www.who.int/emergencies/zika-virus/situation-report/31-march-2016/en/> Acesso em 11 mai. 2017.

[8] IBGE Censo demográfico de Cascavel-PR 2010. Disponivel em: <http://cidades.ibge.gov.br/xtras/temas.php?lang=&codmun=410480&idtema=1&search=parana|cascavel|censo-demografico-2010:-sinopse-> Acesso em 11 mai. 2017.

[9] PALMER, S. R. FELSING, J. M. A Practical Guide to Feature Driven Development. The Coad Series. New Jersey USA: Prentice Hall, 2002.

[10] Sommerville, I. (2003). Engenharia de Software. São Paulo: Addison Wesley.

[11] Sommerville, I.; Kotonya, G. (1997). Requirements Engineering. New York: J. Wiley & Sons.

[12] Pressmann, R. (1995). Engenharia de Software. São Paulo: Makron Books.

[13] PostgreSQL. Disponivel em: <http://www.postgresql.org/> Acesso em 11 mai. 2017.

[14] PostGis. Spatial and Geographic objects for postgreSQL. Disponivel em: <http://postgis.net/> acesso em 11 mai. 2017.

[15] JAVA. Software. Oracle. Disponivel em: <http://www.oracle.com/br/java/overview/index.html > Acesso em 11 mai. 2017.

[16] JavaScript. JavaScript Tutorial. Disponivel em: <http://www.w3schools.com/js/> Acesso em 11 mai. 2017.

[17] JQuery. jQuery foundation. Disponivel em: <https://jquery.org/> Acesso em 11 mai. 2017.

[18] Openlayers. A high-performance, feature-packed library for all you mapping needs. Disponivel em: <https://openlayers.org/> acesso em 11 mai. 2017.

[19] HTML5. HTML5 Introduction. Disponivel em: <http://www.w3schools.com/html/html5_intro.asp> Acesso em 11 mai. 2017.

[20] CSS. CSS Tutorial. Disponivel em: <http://www.w3schools.com/css/> Acesso em 11 mai. 2017.

[21] JSON. Introducin JSON. Disponivel em: <http://www.json.org/> Acesso em 11 mai. 2017.

[22] PLAY. Play. Disponivel em: <https://www.playframework.com/> Acesso em 11 mai. 2017.

[23] Bootstrap. Bootstrap 3 tutorial. Disponivel em: <http://www.w3schools.com/bootstrap/> Acesso em 11 mai. 2017.

[24] XML. Introduction to XML. Disponivel em: <http://www.w3schools.com/xml/xml_whatis.asp> Acesso em 11 mai. 2017.

[25] SQLite. Small. Fast. Reliable. Choose any three. Disponível em: https://www.sqlite.org/ Acesso em 18 ago. 2017.

[26] GOMES, J.M.; VELHO, L. Computação Visual: Imagens. Rio, SBM, 1995.

[27] CÂMARA, G., DAVIS, C., MONTEIRO, A. M. V. Intrudução à Ciência da Geoinformação – INPE. São José dos Pinhais, 2001.

[28] MEDEIROS, Anderson. Geoprocessamento e suas tecnologias parte 1.disponivel em: <http://www.andersonmedeiros.com/geotecnologias-parte1/> acesso em: 3 de julho de 2017.

[29] QGIS. A free and Open Source Geographic Information System. Disponível em <http://www.qgis.org/en/site/> Acesso em : 18 ago. 2017

[30] Shapefile. ESRI Shapefile Technical Description. Disponível em: <https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf> Acesso em: 18 ago. 2017.

[31] Portal da Saúde. Sistemas de Informação. Disponível em: http://portal.saude.gov.br/portal/saude/visualizar_texto.cfm?idtxt=21383. Acesso em: 07 mar. 2013.

[32] Rozendaal J. A. Vector Control - Methods for Use by Individuals and Communities. World Health Organization: Geneva, 1997.

[33] Ministério da Saúde (MS). Diagnóstico rápido nos municípios para vigilância entomológica do Aedes Aegypti no Brasil (LIRAa). MS: Brasília, 2015. 62 p.

[34] World Health Organization. Dengue and severe dengue. Disponível em: <http://www.who.int/mediacentre/factsheets/fs117/en/>. Acesso em 16 dez. 2013.

[35] DengueME: A Tool for the Modeling and Simulation of Dengue Spatiotemporal Dynamics. Disponível em: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5036753/> Acesso em: 18 ago. 2017.

[36] Telessaúde. Núcleo Interestadual RN-PB. Disponível em : <http://telessaude.ufrn.br/index.php/sobre-o-programa > Acesso em: 18 ago. 2017.

[37] Observatório Nacional da Dengue. O Projeto. Disponível em : <http://www.observatorio.inweb.org.br/dengue/conteudo/sobre > Acesso em : 18 ago. 2017.

[38] ECOVEC. Conexões para um mundo sem epidemias. Disponível em: <http://www.midengue.com.br/clientes/midengue.php> Acesso em: 18 ago. 2017.

[39] DengueNet. World Health Organization. Disponível em: <http://apps.who.int/globalatlas/default.asp>. Acesso em 13 fev. 2014.

[40] TWITTER. Disponivel em: <http://www.twitter.com>. Acesso em 17 de jul. 2017.

[41] INFO DENGUE. Disponivel em: < https://info.dengue.mat.br/>. Acesso em 17 de jul. 2017.

[42] Dengue Trends. Disponivel em: < https://www.google.org/flutrends/about/>. Acesso em 17 de jul. 2017.

[43] Governo do Rio de Janeiro. Monitora Dengue. Rio de Janeiro, 2015.

[44] NEGREIROS, Marcos José et al . Integração de sistemas computacionais e modelos logísticos de otimização para prevenção e combate à dengue. Pesqui. Oper., Rio de Janeiro , v. 28, n. 1, p. 1-27, Apr. 2008 . Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0101-74382008000100001&lng=en&nrm=iso>. Acesso em 19 Ago. 2017.

[45] RIZZI, Claudia Brandelero et al. SIGDENGUE: Um Sistema de Informação para o Acompanhamento e Gestão de Ações sobre Dengue com Enfoque às Atividades de Notificação, Raio e Bloqueio. iSys - Revista Brasileira de Sistemas de Informação, [S.l.], v. 9, n. 1, p. 101-117, may 2016. ISSN 1984-2902. Disponível em: <http://www.seer.unirio.br/index.php/isys/article/view/5282>. Acesso em: 19 aug. 2017.

[46] Pencil Project. An open-source GUI prototyping tool that's available for ALL platforms. Disponível em: < https://pencil.evolus.vn/> Acesso em: 19 ago. 2017.

[47] helloacm. Model View Controller explained in C. Disponível em: <https://helloacm.com/model-view-controller-explained-in-c> Acesso em: 27 nov. 2017.

[48] Open Street Maps. Disponível em: <https://www.openstreetmap.org/> Acesso em: 27 nov. 2017.