



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA CENTRAL
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

FELIPE DE SOUSA SILVA

**AGIE 2.0: UM SISTEMA DE GERENCIAMENTO DE ESTACIONAMENTO COM
ROTAS DE ORIENTAÇÃO AO USUÁRIO**

TERESINA, PIAUÍ

2021

FELIPE DE SOUSA SILVA

AGIE 2.0: UM SISTEMA DE GERENCIAMENTO DE ESTACIONAMENTO COM ROTAS
DE ORIENTAÇÃO AO USUÁRIO

Projeto apresentado à Banca Examinadora como requisito para aprovação na disciplina de Trabalho de Conclusão de Curso do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí.

Orientador: Prof. M^e Valéria Oliveira Costa

TERESINA, PIAUÍ

2021

Dados Internacionais de Catalogação na Publicação (CIP) de acordo com ISBD

Silva, Felipe de Sousa

S586a AGIE 2.0 : um sistema de gerenciamento de estacionamento com rotas de orientação ao usuário / Felipe de Sousa Silva. - 2021.
34 f.: il. color.

Trabalho de Conclusão de Curso (Tecnologia em Análise e Desenvolvimento de Sistemas) - Instituto Federal do Piauí, Campus Teresina Central, 2021.

Orientadora : Profa. Ma. Valéria Oliveira Costa.

1. Estacionamento inteligente. 2. Rotas. 3. Roteamento. 4. Georreferenciamento. 5. Mobilidade urbana. I. Título.

CDD - 004.21

Elaborado por Sindya Santos Melo CRB 3/1085

FELIPE DE SOUSA SILVA

AGIE 2.0: UM SISTEMA DE GERENCIAMENTO DE ESTACIONAMENTO COM ROTAS
DE ORIENTAÇÃO AO USUÁRIO

Projeto apresentado à Banca Examinadora como requisito para aprovação na disciplina de Trabalho de Conclusão de Curso do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas do Instituto Federal de Educação, Ciência e Tecnologia do Piauí.

Aprovado pela banca examinadora em _____.

BANCA EXAMINADORA:

Prof. M^e Valéria Oliveira Costa

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

Prof. M^e Eduilson Livio Neves Da Costa Carneiro

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

Prof. Dr. Ricardo Martins Ramos

Instituto Federal de Educação, Ciência e Tecnologia do Piauí - IFPI

TERESINA, PIAUÍ

2021

RESUMO

O grande aumento no volume de vendas de veículos, somado à falta de infraestrutura, tem gerado grandes dificuldades relacionadas à mobilidade urbana. O presente trabalho propõe uma solução para um dos problemas relacionados à mobilidade urbana: encontrar vagas de estacionamento. O sistema permite aos usuários a visualização, em tempo real, das vagas disponíveis de um determinado estacionamento em um mapa interativo. Além disso, é possível obter a localização do usuário e traçar uma rota dentro do estacionamento, com o objetivo de guiá-lo até a vaga escolhida.

Palavras-chaves: estacionamento inteligente, rotas, roteamento, georreferenciamento, mobilidade urbana.

ABSTRACT

The large increase in the volume of vehicle sales, coupled with the lack of infrastructure, has generated great difficulties related to urban mobility. This paper proposes a solution to one of the problems related to urban mobility: finding parking spaces. The application allows the real time visualization of available parking spaces on a interactive map. In addition, it's possible to obtain the user's location and route within the parking lot, to guide him to the chosen parking spot.

Key-words: Smart parking, routes, routing, georeferencing, urban mobility.

LISTA DE ILUSTRAÇÕES

Figura 1 – Latitude	16
Figura 2 – Longitude	16
Figura 3 – Arquitetura AGIE 2.0	23
Figura 4 – Diagrama de casos de uso	24
Figura 5 – Cadastro do cliente	26
Figura 6 – Exemplo de imagem de fundo do estacionamento	28
Figura 7 – Gerenciamento dos arquivos geométricos	28
Figura 8 – Mapa interativo das vagas	29
Figura 9 – Visualizar vagas	29
Figura 10 – Exibir rota - entrada	30
Figura 11 – Exibir rota - vaga	30
Figura 12 – Rota da vaga marcada	31
Figura 13 – Verificação de intenção de saída	31

LISTA DE ABREVIATURAS E SIGLAS

DENATRAN	Departamento Nacional de Trânsito
IFPI	Instituto Federal de Educação, Ciência e Tecnologia do Piauí
PcD	Pessoas com Deficiência
MVC	Modelo, Visão, Controlador
SIG	Sistemas de Informações Geográficas
GPS	Sistema de Posicionamento Global
GML	Linguagem de Marcação Geográfica
OGC	Consórcio Geoespacial Aberto
BLE	<i>Bluetooth Low Energy</i>
RoR	Ruby on Rails
TCC	Trabalho de Conclusão de Curso
W3C	World Wide Web Consortium
ESRI	Environmental Systems Research Institute
GIS	Geographic Information System

SUMÁRIO

1	INTRODUÇÃO	10
1.1	Contextualização	10
1.2	Justificativa	10
1.3	Objetivos	11
1.3.1	Objetivo Geral	11
1.3.2	Objetivos Específicos	11
2	FUNDAMENTAÇÃO TEÓRICA	12
2.1	Arquitetura de software	12
2.1.1	O padrão MVC	12
2.1.2	Linguagem de programação e Framework	13
2.2	Geoprocessamento	13
2.2.1	Dados geográficos	13
2.2.2	Georreferenciamento	14
2.2.3	Shapefiles	14
2.2.4	Banco de Dados Geográficos	15
2.2.5	API de Geolocalização	15
2.2.6	Posicionamento	15
2.2.7	Sistema de Coordenadas Geográficas	16
2.2.8	Sistema Geodésico de Referência	16
2.2.9	Representação de objetos georreferenciados	17
3	TRABALHOS RELACIONADOS	18
3.1	AGIE - SISTEMA PARA GESTÃO DE ESTACIONAMENTOS	18
3.2	Implementation of Smart Parking Guidance System based on Parking Lots Sensors Networks	18
3.3	Smart Parking app - um aplicativo móvel para visualização de vagas em Estacionamento Inteligente	19
4	METODOLOGIA	20
4.1	Levantamento de requisitos	20
4.1.1	Requisitos para desenvolvimento do mapa	20
4.2	Ferramentas e métodos	20
4.2.1	Documentação	20
4.3	Linguagem e <i>framework</i>	21
4.3.1	Sistema Servidor	21

4.3.2	Sistema Cliente	21
5	AGIE 2.0	22
5.1	Arquitetura	22
5.1.1	Detalhamento da arquitetura	22
5.2	Diagrama de Casos de Uso	24
6	RESULTADOS	26
6.1	Servidor	26
6.2	Cliente	29
7	CONCLUSÃO E TRABALHOS FUTUROS	32
	REFERÊNCIAS	33

1 INTRODUÇÃO

Neste capítulo será realizada uma contextualização do problema, apresentando a justificativa para o desenvolvimento do presente trabalho. Em seguida, serão descritos os objetivos geral e específicos.

1.1 CONTEXTUALIZAÇÃO

A frota mundial de veículos cresce a cada ano. Pesquisas apontam que o Brasil ultrapassou a marca de 57 milhões de automóveis (DENATRAN, 2020), o que corresponde a aproximadamente um veículo a cada quatro habitantes, se considerarmos a projeção da população brasileira para o ano de 2021, que é de aproximadamente 212 milhões de habitantes (IBGE, 2021). De toda a frota nacional, o Piauí contribui com mais de 391 mil veículos para essa estatística (DENATRAN, 2020).

Com o rápido crescimento da quantidade de veículos, há também, consequentemente, um aumento na busca por locais para estacionar. Devido à baixa oferta de estacionamentos nas cidades brasileiras comparada à quantidade de veículos, essa busca tende a tornar-se uma dor de cabeça para os donos de automóveis.

Além do tempo perdido, as longas buscas por vagas de estacionamento geram prejuízo financeiro para os motoristas, visto que mais tempo buscando uma vaga representa maior desperdício de combustível. Além dos prejuízos já citados, a emissão de poluentes acaba por ser maior, prejudicando diretamente o meio ambiente.

Junto ao desperdício de recursos na busca por uma vaga, podemos citar ainda um outro problema recorrente: a falta de sinalização dentro de estacionamentos. Muitos estacionamentos não dispõem de uma forma eficiente para informar sobre a disponibilidade de vagas. É comum ver pessoas em frente a grandes estacionamentos indicando sobre a disponibilidade de vagas, mas sem indicações sobre quais vagas estão disponíveis ou de como chegar até a vaga desejada. Esta falta de informações acaba por gerar perdas de tempo para quem procura uma vaga para estacionar.

1.2 JUSTIFICATIVA

Devido ao crescimento acelerado da frota de veículos, a alta procura por vagas de estacionamentos associada à precária infraestrutura urbana das cidades brasileiras gera a necessidade da busca por soluções inovadoras e eficientes para resolução desse problema. Com isso, o conceito de estacionamentos inteligentes tem se tornado cada vez mais presente em estratégias para solução de problemas de mobilidade urbana.

Para amenizar problemas causados pela precária infraestrutura, é importante a disponibilização de informações referentes à disponibilidade de vagas, evitando o desperdício de recursos em uma longa busca por estacionamento com vagas disponíveis.

Além dos problemas relacionados à busca por estacionamento, encontrar uma vaga específica dentro do estacionamento nem sempre é uma tarefa fácil. Pessoas com um fraco senso de direção tendem a se perder mais facilmente, e enfrentar dificuldades em encontrar o seu destino (PADGITT; HUND, 2012).

Considerando os problemas descritos anteriormente, seria de grande utilidade pública uma ferramenta para mostrar, em tempo real, a disponibilidade das vagas para os usuários do estacionamento, bem como informações relativas à quantidade exata de vagas disponíveis. Além disso, uma solução para orientação através de um mapa pode contribuir de forma positiva para o dia-a-dia dos usuários. A orientação deve funcionar de forma semelhante à funcionalidade oferecida pelo Google Maps, com rotas até o destino, mas para o ambiente interno do estacionamento.

O trabalho aqui descrito propõe uma solução aplicada ao estacionamento do Campus Teresina Central do Instituto Federal do Piauí, mas escalável, podendo ser aplicado a quaisquer estacionamentos que sejam mapeados nos padrões a serem exigidos pelo sistema.

1.3 OBJETIVOS

1.3.1 Objetivo Geral

Desenvolver um sistema para auxiliar o usuário no processo de busca por vagas livres no estacionamento, reduzindo o tempo gasto para encontrar uma vaga específica.

1.3.2 Objetivos Específicos

- Apresentar, em um mapa interativo, as vagas disponíveis e ocupadas;
- Auxiliar o usuário do estacionamento na localização da vaga, exibindo uma rota até a vaga escolhida;
- Orientar o usuário no retorno ao seu veículo, mostrando o caminho até o local em que o veículo se encontra estacionado.
- Oferecer uma comunicação entre os usuários, permitindo que um usuário informe sobre a sua saída e liberação da vaga ocupada.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão apresentados alguns conceitos relevantes relacionados a este trabalho, dentre os quais podemos destacar os seguintes itens diretamente relacionados ao desenvolvimento de software: padrões de arquitetura de *software*, linguagem de programação e *framework*. Devido à sua multidisciplinaridade, o presente trabalho engloba também conceitos relacionados ao geoprocessamento: dados geográficos, georreferenciamento, coordenadas geográficas, navegação e posicionamento, API de geolocalização e sistema geodésico de referência.

2.1 ARQUITETURA DE SOFTWARE

O desenvolvimento profissional de software requer um processo adequado para que o resultado seja um produto de qualidade. A não utilização de um processo adequado na construção de *software* pode trazer algumas consequências, como estouro de cronograma e orçamento, produto final de baixa qualidade ou produtos com dificuldades de evolução e manutenção (DIJKSTRA, 1972). Para evitar esses e outros problemas, faz-se uso das técnicas e conceitos de engenharia de *software*, seguindo uma metodologia de desenvolvimento e gerando artefatos de documentação.

A arquitetura de software tem o papel fundamental de descrever a estrutura geral do sistema, identificar os principais componentes estruturais e como se relacionam, bem como a distribuição desses componentes (SOMMERVILLE, 2011).

De acordo com Bass, Clements e Kazman (2012), elementos arquiteturais são, em alguns casos, compostos de formas específicas a fim de solucionar problemas específicos. Estas composições são chamadas de padrões arquiteturais. Entre os padrões citados por Bass, Clements e Kazman (2012), o presente trabalho utiliza os seguintes:

- Model-view-controller (MVC): no padrão de arquitetura MVC, o sistema é dividido em três componentes: modelo, visão e controlador
- Padrão cliente-servidor: nesse padrão, os componentes são os clientes e os servidores. O principal tipo de conector é um conector de dados conduzido por um protocolo de requisição/resposta usado pelo cliente para solicitar serviços ao servidor

2.1.1 O padrão MVC

Criado em 1979 por Trygve Reenskaug (RUBY; THOMAS et al., 2009), o MVC consiste em três tipos de objetos:

- *Model* (modelo): componente responsável pelo processamento dos dados e comunicação com o banco de dados. O modelo é também responsável pelas regras de

negócio do sistema.

- *View* (visão): é a interface visual disponibilizada para o usuário. A *view* especifica como o modelo será exibido.
- *Controller* (controlador): responsável por realizar a troca de dados com a visão. O controlador comunica-se com o modelo para executar tarefas determinadas pela regra de negócio.

2.1.2 Linguagem de programação e Framework

Para auxiliar no desenvolvimento de programas, pode-se fazer uso de um *framework*, reduzindo assim parte do processo manual de configurações da estrutura da aplicação. O *framework* determina a arquitetura do projeto, a forma como é organizada a estrutura da aplicação (GAMMA, 1995).

Este trabalho utiliza os seguintes *frameworks*:

- Ruby on Rails (RoR): gratuito, de código aberto, que permite o uso da linguagem Ruby para desenvolver aplicações *web*. O RoR conta com uma grande diversidade de ferramentas para auxiliar o desenvolvimento, chamadas de *gems*. O RoR utiliza o padrão de arquitetura MVC (BÄCHLE; KIRCHBERG, 2007).
- Angular: um poderoso *framework* de código aberto para desenvolvimento de aplicações *web*, utilizando *JavaScript*, criada por desenvolvedores da Google (FREEMAN, 2018).
- Ionic: *framework* de código aberto para desenvolvimento de aplicações para dispositivos móveis, bem como para computadores, utilizando tecnologias *web* (IONIC, 2019).

2.2 GEOPROCESSAMENTO

O termo geoprocessamento pode ser definido como uma disciplina do conhecimento que trata de informações geográficas, com uso da matemática e da computação (DAVIS; CÂMARA; MONTEIRO, 2001). Geoprocessamento pode também ser definido como um conjunto de tecnologias utilizadas para coleta e processamento de informações espaciais, bem como desenvolvimento de sistemas que utilizem dados geográficos (ROSA; BRITO, 1996).

2.2.1 Dados geográficos

Os dados geográficos são tratados computacionalmente por Sistemas de Informações Geográficas (SIG). Um SIG difere de um sistema de informação convencional pelo fato de ser capaz de armazenar as geometrias de diferentes tipos de dados geográficos, além de atributos referentes às informações semânticas (CASANOVA et al., 2005). De acordo com Rosa e

BRITO (1996), um SIG é destinado a realizar as tarefas relacionadas ao processamento de dados geográficos, incluindo armazenamento e apresentação desses dados.

De acordo com Rosa e BRITO (1996), os dados geográficos de um SIG podem ser representados por:

- Pontos: elementos compostos por um único par de coordenadas x , y e que estão presentes em todos os outros tipos de representação.
- Linhas: elementos compostos por um par de pontos.
- Áreas ou Polígonos: as áreas são formadas por um conjunto de pontos interligados, com objetivo de descrever as propriedades topológicas de um objeto geográfico.

2.2.2 Georreferenciamento

O georreferenciamento pode ser visto como um mapeamento de dados geográficos de objetos, construções ou áreas, para um sistema de coordenadas conhecido, para que possa ser visualizado e analisado com outros dados geográficos (WADE; SOMMER et al., 2006).

Os dados de objetos georreferenciados possuem, entre outros atributos, valores referentes à sua localização geográfica a partir de um sistema de coordenadas. Esses objetos são chamados de objetos geográficos, devido à possibilidade de serem relacionados a uma localização geográfica (HACKELOEER et al., 2014).

De acordo com Hackeloeer et al. (2014), objetos geográficos possuem três tipos distintos de informações, que podem ser usados para referenciá-los:

- Informações geométricas: descrevem as propriedades geométricas de um objeto geográfico. Estes objetos são utilizados para representar o formato geométrico do objeto geográfico, como, por exemplo, o formato de uma área de estacionamento.
- Informações topológicas: lidam com propriedades que são preservadas sob deformações contínuas dos objetos. Estas informações podem representar o relacionamento espacial entre objetos geográficos. Os pontos que ligam as linhas de uma rota são exemplos de informações topológicas.
- Informações semânticas: incluem as propriedades semânticas dos objetos geográficos, como, por exemplo, nome de um estacionamento, bem como informações que podem ser usadas para identificar um objeto de forma única.

2.2.3 Shapefiles

Os shapefiles são uma coleção de arquivos de diferentes extensões, utilizados para bases de dados geoespaciais e vetoriais em sistemas de informação geográfica (do inglês, GIS - Geographic Information System), desenvolvidos, definidos e regulados pela ESRI (1998) (Environmental Systems Research Institute).

Os arquivos shapefiles, que podem ser de diversas extensões, devem ser armazenados no mesmo diretório ArcGIS (2011). Entre as extensões disponíveis, o AGIE 2.0 utiliza as seguintes:

- shp: é o principal arquivo, que armazena as características da geometria propriamente dita;
- shx: arquivo utilizado para indexar as características da geometria, para buscas mais rápidas;
- dbf: formato dos atributos, apresentados em colunas para cada objeto geométrico;
- prj: arquivo que indica o sistema de coordenadas e projeção utilizado, ou seja, qual sistema geodésico de referência será utilizado.

2.2.4 Banco de Dados Geográficos

Um banco de dados é utilizado para armazenar todos os dados que necessitem ser persistidos. Um banco de dados geográficos difere dos demais bancos de dados pela sua capacidade de armazenar todos os dados geométricos de um objeto geográfico, além dos atributos descritivos (ROSA; BRITO, 1996).

Além da capacidade de armazenar e disponibilizar dados das geometrias dos objetos, um banco de dados geográfico possui operações específicas para manipulação dos dados geométricos (ROSA; BRITO, 1996). Como exemplo de operações específicas dos bancos de dados geográficos, podemos citar os algoritmos de roteamento que calculam uma rota entre dois pontos. O PostgreSQL é um exemplo de banco de dados com suporte ao tratamento de dados geográficos.

2.2.5 API de Geolocalização

Em sistemas que utilizam o modelo de arquitetura cliente-servidor, é comum a utilização de Interface de Programação de Aplicativos (do inglês *Application Programming Interface*, API). Uma API expõe um conjunto de dados e operações relacionadas aos recursos disponíveis em um sistema, facilitando a troca de dados entre programas (cliente e servidor) (MASSE, 2011).

Dentro de uma variedade de tipos de APIs, existem as APIs de geolocalização que são utilizadas por aplicações que trabalham com mapas e objetos georreferenciados, para obter informações de localização e posicionamento de um determinado dispositivo eletrônico. A API de geolocalização utilizada no presente trabalho é especificada pelo W3C (2016).

2.2.6 Posicionamento

O Sistema de Posicionamento Global (GPS) é uma das tecnologias mais utilizadas quando se trata de posicionamento e localização. O sistema é composto por 24 satélites distribuídos ao redor do planeta, que enviam dados para os equipamentos receptores, permitindo obter a localização do receptor em um plano tridimensional. Para isso, é necessária uma conectividade com pelo menos quatro satélites (HASEGAWA et al., 1999).

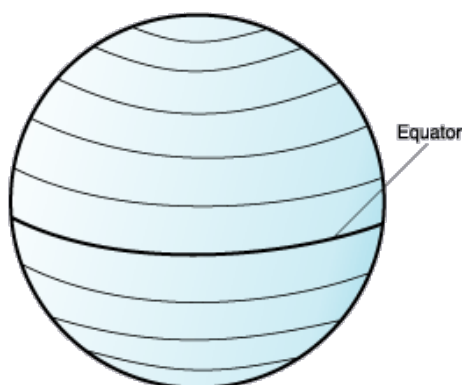
Para exibir o posicionamento obtido por um receptor ou dados geométricos armazenados em um banco de dados, utiliza-se um sistema de coordenadas para a localização da geometria. O sistema mais comumente utilizado para a superfície da Terra usa coordenadas compostas por latitude e longitude (HASEGAWA et al., 1999). O trabalho descrito no presente documento adotou o uso do sistema de coordenadas geográficas, que utiliza latitude e longitude para referenciar um ponto na superfície terrestre.

2.2.7 Sistema de Coordenadas Geográficas

Sistema de coordenadas geográficas é um sistema que utiliza medidas esféricas de longitude e latitude para descrever a localização geográfica de um determinado ponto na superfície da Terra. Qualquer local na Terra pode ser representado como um ponto, formado por coordenadas de longitude e latitude (IBM, 2016).

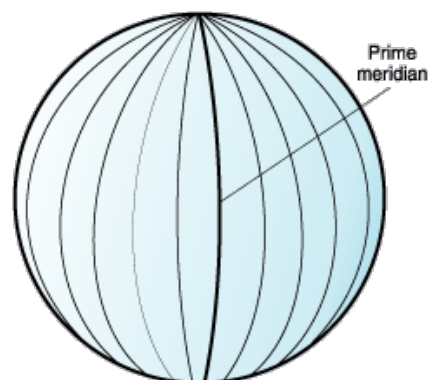
As coordenadas de longitude são os meridianos, linhas perpendiculares ao Equador, enquanto as coordenadas de latitude são os paralelos, que são as linhas paralelas à linha do Equador. As coordenadas são comumente representadas em graus, minutos e segundos (IBM, 2016).

Figura 1 – Latitude



Fonte: IBM (2016)

Figura 2 – Longitude



Fonte: IBM (2016)

2.2.8 Sistema Geodésico de Referência

Um sistema geodésico de referência é um sistema composto por representações geométricas da superfície da Terra, posicionada no espaço, utilizado para realizar a localização dos pontos de coordenadas na superfície terrestre, de forma única (IBGE, 2019).

Entre os padrões de sistemas geodésicos existentes, optou-se por utilizar o World Geodetic System 1984 (WGS 84), criado pelo Departamento de Defesa dos Estados Unidos em 1984 (KUMAR, 1988). O WGS 84 é um padrão que define as referências de coordenadas para estabelecer latitude e longitude, além do posicionamento e alturas de navegação (NGA, 2014).

Um sistema geodésico de referência é importante para determinar as coordenadas e representá-las de forma adequada, como as de latitude e longitude.

2.2.9 Representação de objetos georreferenciados

Para que o usuário da aplicação possa visualizar os dados geográficos dos objetos georreferenciados armazenados no banco de dados, podemos representá-los em um componente de mapa em uma aplicação, seja *web* ou para dispositivos móveis.

Este trabalho optou por utilizar a Google Maps API, uma biblioteca para desenvolvimento de mapas. Esta biblioteca permite a representação de objetos georreferenciados em um mapa, com linhas, pontos, polígonos e imagens (GOOGLE, 2019).

3 TRABALHOS RELACIONADOS

Este capítulo apresenta trabalhos cujo conteúdo está relacionado ao tema escolhido, como monitoramento de vagas de estacionamento com uso de equipamentos eletrônicos, ou posicionamento e orientação geográfica como formas de auxílio ao usuário.

3.1 AGIE - SISTEMA PARA GESTÃO DE ESTACIONAMENTOS

A autora propõe um sistema para gestão de estacionamentos com recursos como: visualização da disponibilidade de cada vaga monitorada, análises de estatísticas referentes ao uso do estacionamento e alertas emitidos pelos administradores do sistema (LOPES, 2019).

O sistema desenvolvido recebe dados captados através de um sensor acoplado a cada uma das vagas do estacionamento, e dispõe de uma interface *web* para apresentar dados estatísticos referentes ao uso das vagas, bem como a disponibilidade das mesmas (LOPES, 2019).

Através da sua interface *web*, o sistema AGIE informa se determinada vaga está livre ou ocupada (LOPES, 2019). Além de exibir um mapa do estacionamento, o sistema AGIE poderia oferecer recursos para a navegação e orientação do usuário no ambiente do estacionamento, indicando sua posição no mapa, bem como sugerir uma rota da localização atual do usuário até a vaga selecionada.

O trabalho descrito neste documento é uma evolução do sistema descrito acima, visando a continuidade do desenvolvimento do projeto AGIE, e utiliza dados disponibilizados pela autora.

3.2 IMPLEMENTATION OF SMART PARKING GUIDANCE SYSTEM BASED ON PARKING LOTS SENSORS NETWORKS

O trabalho apresentado por Cai, Zhang e Pan (2015) propõe um sistema de orientação para estacionamentos. O sistema consiste em uma rede de sensores para as vagas, nó coletor e uma aplicação para gerenciar o estacionamento.

Como descrito por Cai, Zhang e Pan (2015), a rede de sensores ultrassônicos realiza a detecção dos veículos nas vagas do estacionamento. O autor utiliza protocolo de comunicação ZigBee para enviar as informações sobre as vagas para o nó coletor que, por sua vez, envia tais informações para o sistema de gerenciamento do estacionamento.

A ferramenta proposta exibe informações sobre a disponibilidade das vagas em uma aplicação, bem como sugere uma rota até as vagas do estacionamento. No entanto, o trabalho desenvolvido por Cai, Zhang e Pan (2015) poderia dispor de uma versão do sistema para dispositivos móveis.

3.3 SMART PARKING APP - UM APLICATIVO MÓVEL PARA VISUALIZAÇÃO DE VAGAS EM ESTACIONAMENTO INTELIGENTE

A proposta apresentada por Aoki et al. (2020) trata do desenvolvimento de uma aplicação para visualização de vagas em um estacionamento inteligente, aplicado ao contexto da Universidade Estadual de Campinas (Unicamp).

O sistema desenvolvido consiste em uma aplicação que conta com uma visualização das vagas do estacionamento do campus da Unicamp, além de um cadastro de eventos, onde o usuário pode cadastrar a sua jornada diária, indicando os locais a serem frequentados. A partir daí, o sistema exibe as vagas disponíveis nas proximidades dos locais indicados no cadastro dos eventos.

O trabalho desenvolvido por Aoki et al. (2020) mostrou-se de grande utilidade. Considerando que o sistema já conta com o uso do *Google Maps*, seria de grande valia a adição de um mapa interno do campus, a fim de exibir a localização exata das vagas, dando uma melhor orientação ao usuário.

4 METODOLOGIA

Este capítulo apresenta a metodologia escolhida para o desenvolvimento do projeto, descrevendo as etapas e técnicas utilizadas.

4.1 LEVANTAMENTO DE REQUISITOS

Considerando o *campus* Teresina Central do IFPI como piloto, foram analisadas as demandas já apuradas no trabalho desenvolvido por Lopes (2019), onde são explicitados os problemas enfrentados pelos servidores do instituto em questão. Além disso, foram feitas leituras de trabalhos relacionados com o tema aqui trabalhado, a fim de extrair informações relevantes quanto aos problemas relativos ao mapeamento de estacionamentos.

4.1.1 Requisitos para desenvolvimento do mapa

A solução proposta por este trabalho inclui o uso de mapas georreferenciados. Para o desenvolvimento desta solução, foi necessária a participação de um colaborador da área de Geoprocessamento na realização do mapeamento do estacionamento do IFPI, estacionamento piloto para o presente trabalho.

O mapeamento do estacionamento é necessário para possibilitar a implementação de funcionalidades que utilizam dados geográficos. Como exemplo destes tipos de funcionalidades, podemos citar a navegação dos usuários, que utiliza as coordenadas geográficas do usuário para posicioná-lo no mapa e, assim, indicar uma rota até a vaga de destino.

4.2 FERRAMENTAS E MÉTODOS

4.2.1 Documentação

A documentação é parte muito importante de um software. Para sua produção, foram utilizadas as seguintes ferramentas:

- Astah UML: ferramenta para modelagem de diagramas, como os diagramas de classe e de casos de uso. Utilizado para elaboração dos diagramas necessários para representação dos requisitos levantados.
- Trello: uma plataforma para registrar o planejamento e acompanhamento das tarefas, que permite organizar as atividades por prioridade. O Trello é a ferramenta utilizada neste trabalho para documentar o planejamento das metas, organizando as tarefas de forma clara, de acordo com o planejamento semanal.
- Github: repositório utilizado para armazenamento de código e quaisquer tipos de documentação, com suporte a versionamento dos arquivos.

4.3 LINGUAGEM E *FRAMEWORK*

As linguagens e frameworks foram escolhidos considerando a arquitetura híbrida do projeto.

4.3.1 Sistema Servidor

Para o desenvolvimento do sistema servidor, optou-se pela escolha da linguagem de programação Ruby, uma linguagem de programação orientada a objetos. A adoção dessa linguagem deu-se pelo fato de a mesma possuir os recursos necessários para trabalhar com geociências, sintaxe de fácil compreensão, além de possuir uma documentação clara e objetiva.

Considerando a escolha da linguagem Ruby, decidiu-se por utilizar o *framework* Ruby on Rails (RoR) para um desenvolvimento de forma mais eficiente. O RoR é um *framework* que facilita o desenvolvimento e manutenção de aplicações *web* (RUBY; THOMAS et al., 2009).

Entre os motivos que levaram à escolha do RoR, está o fato de que ele utiliza o padrão de arquitetura MVC. Outro ponto importante é a organização da estrutura dos arquivos do sistema: os arquivos são organizados por grupos de funcionalidade, seguindo ainda a estrutura organizacional do padrão MVC (RUBY; THOMAS et al., 2009).

4.3.2 Sistema Cliente

Para desenvolver o cliente, a escolha foi pelo Ionic Framework, um *framework* de código aberto para desenvolvimento de aplicações para dispositivos móveis, bem como para computadores, utilizando tecnologias *web* (IONIC, 2019).

Atualmente, o Ionic possui integração oficial com o Angular, um *framework* de desenvolvimento que utiliza a linguagem de programação JavaScript. Esta parte do trabalho foi desenvolvido utilizando o *framework* Angular, considerando sua ampla integração com o Ionic Framework.

O sistema cliente é responsável pela disponibilização dos recursos inerentes à visualização dos mapas dos estacionamento e navegação, além da autenticação e gerenciamento de perfil pessoal do usuário. A comunicação do cliente com a base de dados ocorre através da API, disponibilizada pelo servidor.

5 AGIE 2.0

Neste capítulo será apresentado o detalhamento da arquitetura do sistema, além do diagrama de casos de uso.

5.1 ARQUITETURA

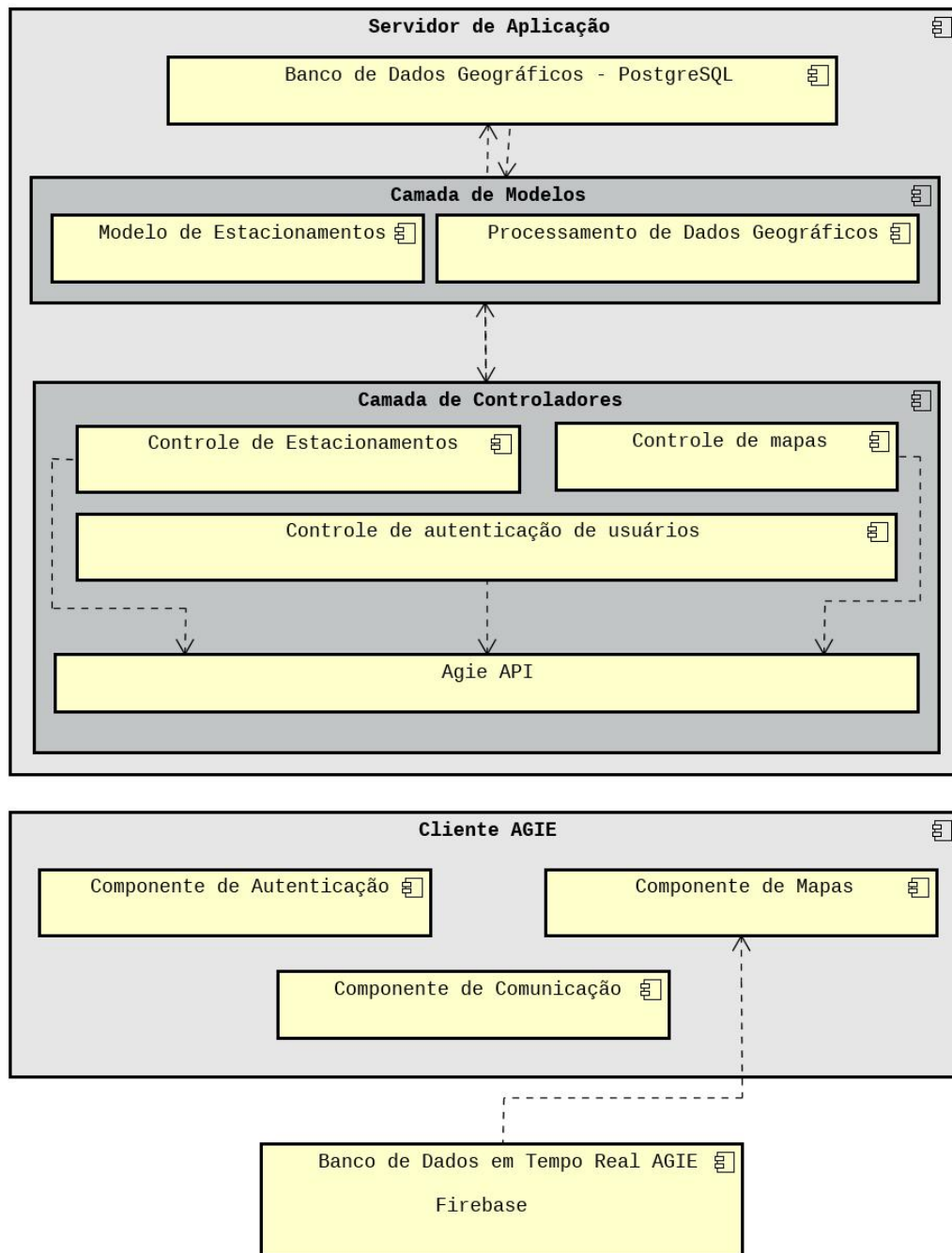
O projeto conta com uma arquitetura híbrida, utilizando o padrão de camadas *Model/View/Controller* (MVC) aliado ao modelo cliente e servidor. Em relação ao padrão MVC, escolha deu-se devido à sua forma de organização que permite trabalhar os componentes de forma separada. Além disso, é o padrão de arquitetura adotado pelo *framework Ruby on Rails* (RoR), escolhido para o desenvolvimento do sistema. Quanto ao modelo cliente e servidor, foi desenvolvida uma aplicação servidor, bem como uma aplicação cliente, devido à demanda de processamento e a necessidade de uma aplicação para dispositivos móveis.

5.1.1 Detalhamento da arquitetura

A arquitetura planejada para o desenvolvimento do sistema, apresentada na Figura 3, está detalhada da seguinte forma:

- **O Servidor de Aplicação AGIE** é responsável por realizar o processamento dos dados, fornecendo retorno através de uma API, aqui chamada de Agie API. O servidor está dividido nos seguintes componentes:
 - Banco de dados geográficos: responsável por armazenar todos os dados que serão utilizados no sistema AGIE. O banco de dados geográficos possui ainda algoritmos de roteamento para realizar o cálculo das rotas de navegação. O sistema utiliza o banco de dados PostgreSQL, que permite trabalhar com dados geográficos.
 - Camada de modelos: possui os modelos responsáveis por realizar o processamento dos dados. O modelo de estacionamento processa os atributos descritivos, como informações de cadastro dos estacionamento. O modelo de processamento de dados geográficos é encarregado por todo o processamento dos dados geométricos dos estacionamentos, bem como coordenadas dos usuários e cálculo de rotas para navegação.
 - Camada de controladores: agrega os componentes responsáveis pela comunicação entre a camada de modelo e o cliente. Através da API, os controladores realizam a disponibilização e coleta de dados referentes à autenticação dos usuários, informações dos estacionamentos e os dados geográficos dos mapas processados, prontos para serem consumidos pela aplicação cliente.

Figura 3 – Arquitetura AGIE 2.0



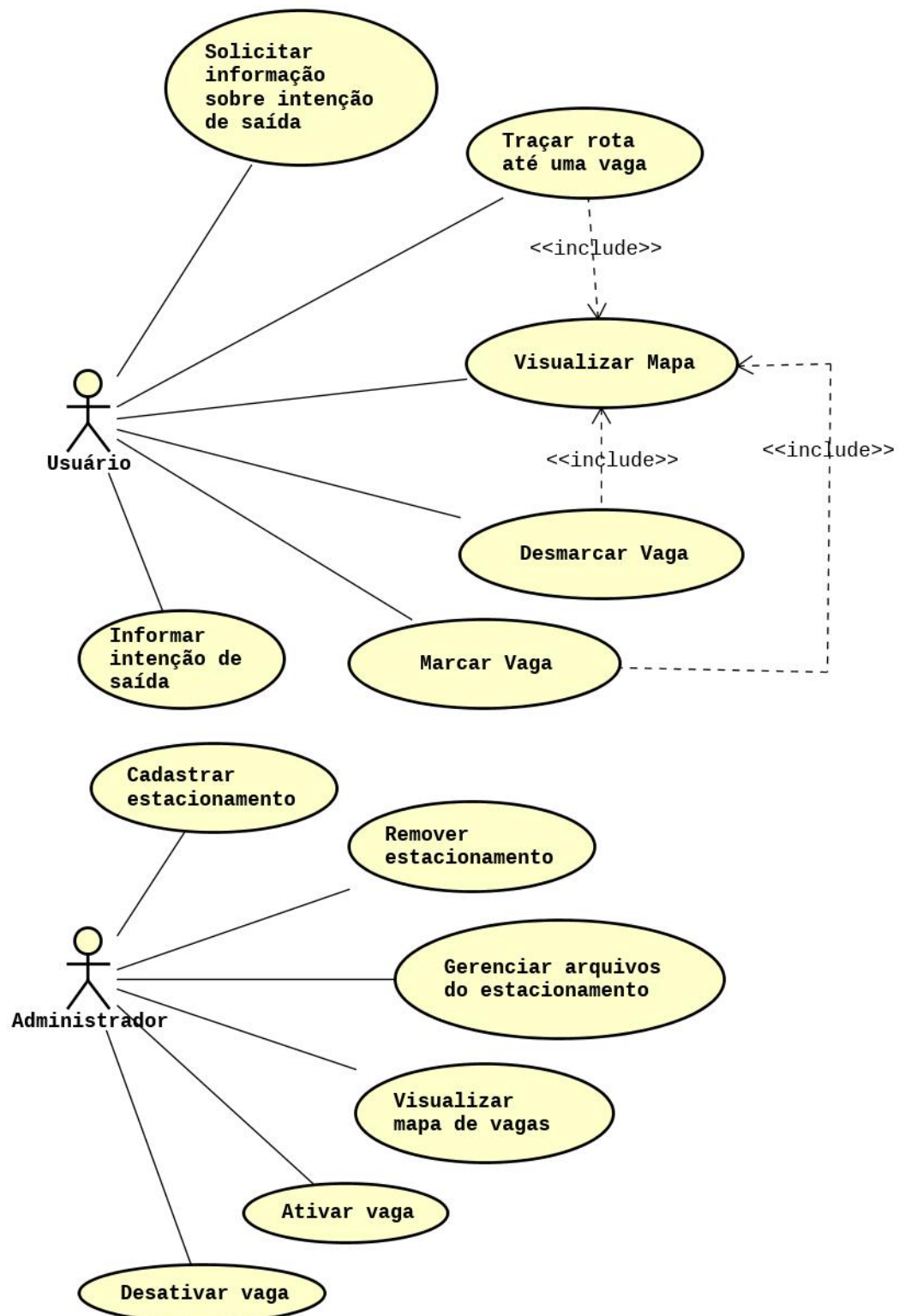
Fonte: feito pelo autor

- O **Cliente AGIE** é a camada visível ao usuário. Aqui encontra-se o componente de autenticação do usuário, requerido para permitir o acesso a recursos do sistema, tais como: visualização das vagas no mapa, rotas no estacionamento, comunicação com outros usuários. O cliente obtém as informações da disponibilidade das vagas diretamente do banco de dados de tempo real do AGIE, no *Firebase*. Para a troca de dados relacionados aos usuários e dados geográficos, que ficam armazenados no banco de dados PostgreSQL, o cliente utiliza a Agie API disponibilizada pelo servidor de aplicação.

5.2 DIAGRAMA DE CASOS DE USO

O diagrama de casos de uso a seguir apresenta as funcionalidades que foram pensadas para este trabalho, a fim de atingir os objetivos propostos neste documento.

Figura 4 – Diagrama de casos de uso



Fonte: feito pelo autor

- Visualizar mapa: o usuário acessa um mapa contendo os estacionamentos cadastrados no sistema, com visualização das vagas correspondentes a cada estacionamento.
- Traçar rota até uma vaga: permite ao usuário selecionar uma vaga, e então o sistema irá exibir a rota mais curta a partir da entrada do estacionamento.
- Marcar vaga: permite ao usuário marcar uma vaga selecionada. É útil para guiar o usuário ao retornar ao estacionamento, auxiliando na busca pela vaga onde seu veículo se encontra estacionado.
- Desmarcar vaga: permite desmarcar uma vaga previamente marcada pelo usuário.
- Solicitar informação sobre intenção de saída: esta funcionalidade permite que o usuário que está aguardando por uma vaga dispare uma notificação para os demais usuários que estiverem no perímetro definido na configuração de cada estacionamento, questionando se alguém que ocupa uma vaga estaria de saída nos próximos minutos.
- Informar intenção de saída: ao receber uma notificação solicitando informações sobre intenção de saída, o usuário poderá manifestar-se informando se pretende sair nos próximos minutos. A resposta é enviada para um *chat* em grupo, visível a qualquer usuário que tenha interesse nas informações sobre intenção de saída.
- Cadastrar estacionamento: funcionalidade exclusiva para os administradores do sistema, permite cadastrar um novo estacionamento, com informações sobre o responsável pelo estacionamento, além da relação de andares, para casos onde seja um edifício de estacionamento.
- Remover estacionamento: permite ao administrador remover um estacionamento do sistema, retirando-o do mapa de estacionamentos.
- Gerenciar arquivos do estacionamento: uma das funcionalidades mais importantes do sistema, permite ao administrador gerenciar todos os arquivos de dados geográficos relacionados a um estacionamento. Entre estes arquivos, podemos citar: arquivo que delimita a área do estacionamento, arquivos de geometrias das vagas e arquivos com a topologia das rotas possíveis entre as vagas.
- Visualizar mapa de vagas: um usuário administrador do sistema acessa a página de visualização das vagas em um mapa interativo.
- Ativar vaga: um usuário administrador, na página de visualização das vagas, seleciona uma vaga atualmente inativa para ativá-la.
- Desativar vaga: um usuário administrador, na página de visualização das vagas, seleciona uma vaga atualmente ativa para desativá-la.

6 RESULTADOS

Este trabalho apresentou o projeto para o desenvolvimento do sistema AGIE 2.0, um sistema para gerenciamento de estacionamento, com suporte a orientação aos usuários. Foram apresentados detalhes relacionados ao planejamento do projeto e ferramentas utilizadas para alcançar os objetivos especificados neste documento.

Como resultado do projeto proposto no presente trabalho, foi realizada uma evolução do sistema AGIE (servidor), incluindo toda a parte de geoprocessamento, necessária para o pleno funcionamento do sistema. Além da evolução do AGIE, este trabalho contempla também o desenvolvimento de uma aplicação para dispositivos móveis (cliente).

6.1 SERVIDOR

Como mencionado anteriormente, foram realizados incrementos ao servidor AGIE. A principal funcionalidade implementada no servidor foi o cadastro de estacionamentos (tratados como clientes dentro da plataforma). O cadastro de estacionamentos, da forma como foi planejado, permite que o sistema seja escalável, além de facilitar o controle do estacionamento.

Ao cadastrar um cliente, pode-se informar um ou mais endereços de estacionamentos, registrando informações relativas aos mesmos, como localização, se o estacionamento possui andares e quais andares deverão ser incluídos no mapa.

Figura 5 – Cadastro do cliente

A interface de administração do sistema AGIE apresenta o seguinte layout para o cadastro de um cliente:

- Barra Superior:** Título "AGIE admin" e informações de usuário/locação.
- Barra Lateral Esquerda:** Menu de navegação com opções como "Início", "Usuários", "Clientes", "Meus estacionamentos", "Informativos", "Ocorrências", "Relatórios" e "Tabelas auxiliares".
- Formulário "Dados do cliente":**
 - CNPJ:** Campo com o valor "11.111.111/1111-11".
 - Razão social:** Campo com o valor "Instituto Federal do Piauí".
 - Nome fantasia:** Campo com o valor "IFPI".
 - Informações de Contato:**
 - Nome:** Campo com o valor "Admin".
 - Telefone fixo:** Campo vazio.
 - Celular:** Campo com o valor "(86) 9999-9999".
 - Botão "Remover contato" (vermelho).
 - Botão "+ Adicionar contato" (verde).
 - Endereços:**
 - CEP:** Campo com o valor "64000-000".
 - Logradouro:** Campo com o valor "Centro".
 - Número:** Campo com o valor "1".
 - Cidade:** Campo com o valor "Teresina".
 - Botão "Remover Endereço" (vermelho).
 - Botão "+ Adicionar endereço" (verde).
 - Andares:**
 - Andar 1:** Campo com o valor "Térreo", checkbox "Contratado" selecionado, botão "Remover Andar" (vermelho).
 - Andar 2:** Campo com o valor "1", checkbox "Contratado" desselecionado, botão "Remover Andar" (vermelho).
 - Botão "+ Adicionar andar" (azul).
 - Botão "+ Adicionar endereço" (verde).

Fonte: feito pelo autor

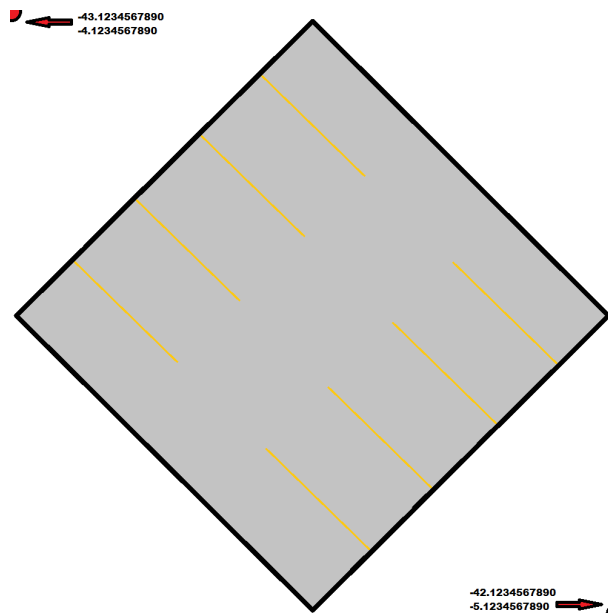
Após realizar o cadastro do cliente, é feito o envio dos arquivos shapefiles do estacionamento, obtidos com o auxílio de um profissional de Geoprocessamento, que realizou o

mapeamento do local e gerou os arquivos geométricos. Os arquivos necessários são:

- Área de estacionamento: deve conter todo o perímetro destinado ao estacionamento. A shapefile deve ser feita em formato de geometria de polígono.
- Áreas importantes: áreas que pertencem ao edifício do estacionamento ou próximas a ele. Arquivos seguem o mesmo padrão da Área de estacionamento.
- Área das vagas individuais: deve conter todo o perímetro destinado a cada vaga individualmente, de maneira a contemplar todos os seus limites. Em um único arquivo shapefile deve conter todas as vagas em polígonos individuais. A shapefile deve ser feita em formato de geometria de multipolígonos e conter um atributo "id", que será o número da vaga.
- Pontos de interesse: devem conter informações de pontos aos arredores ou dentro do próprio estacionamento que tenham informações úteis aos usuários do estacionamento. Todos os pontos de interesses devem estar contemplados em um único arquivo shapefile. A shapefile deve ser feita em formato de multipontos e conter os atributos "id", "nome" e "descricao".
- Nós das rotas: devem conter todos os nós ou pontos de encontros de linhas de rotas possíveis dentro do estacionamento. A shapefile deve ser feita em formato de multipontos e conter um atributo "id", identificador do nó.
- Rotas internas do estacionamento: devem conter as rotas possíveis de serem realizadas. A shapefile deve ser no formato multilinhas, e deve conter os atributos:
 - gid: código identificador único para cada trecho de rota.
 - cost_d: o custo de percurso do trecho em metros. É a dimensão do comprimento do trecho a ser tratado.
 - target: o ponto para o qual o trecho se direciona. Seu valor é compatível com o “Nó das rotas”.
 - source: o ponto no qual o trecho se inicia. Tem valor compatível com o “Nó das rotas”.
 - reverse_co: o custo de percurso do trecho em metros. É o comprimento do trecho a ser tratado.
 - oneway: identificador de direção do sentido naquele trecho da rota, deve ser classificado em 3 tipos: B (bidirecional), FT (unidirecional, partindo de "target" em direção a "source" ou TF (unidirecional, partindo de "source" em direção a "target").

Além dos arquivos shapefiles, é necessário também o envio da imagem de fundo do estacionamento, em formato PNG (Portable Network Graphics), com fundo transparente e em resolução 1000x1000 pixels. A imagem deve conter as coordenadas dos cantos superior esquerdo e inferior direito. As coordenadas informadas devem ser da localização real no mapa.

Figura 6 – Exemplo de imagem de fundo do estacionamento



Fonte: feito pelo autor

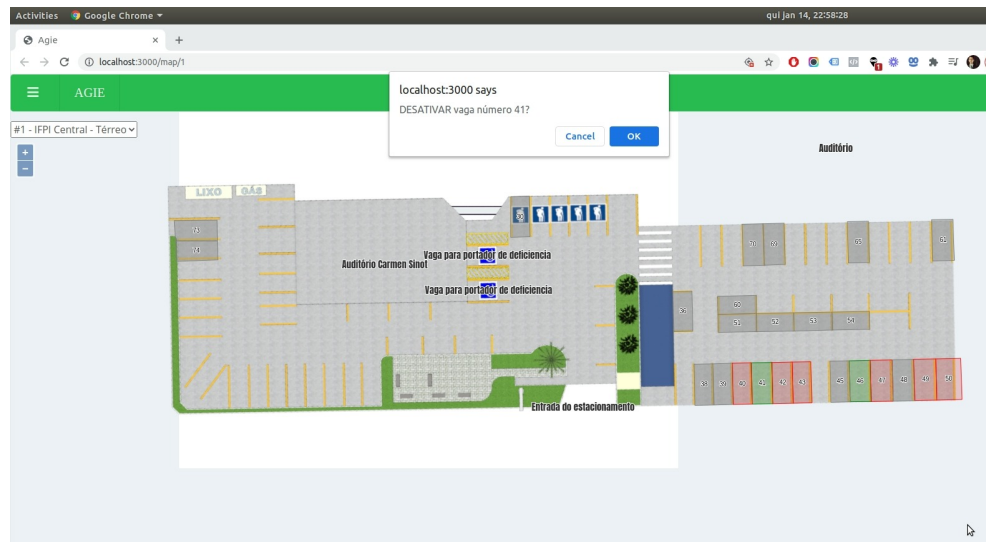
Os arquivos enviados serão processados pelo sistema utilizando a *gem* RGeo, uma ferramenta de código aberto disponível para o framework Ruby on Rails. Por fim, o sistema se encarregará de armazenar os dados processados no banco de dados PostgreSQL.

Figura 7 – Gerenciamento dos arquivos geométricos

Fonte: feito pelo autor

Concluída a etapa de envio dos arquivos, o sistema já está pronto para uso das funcionalidades de monitoramento e gerenciamento das vagas. Uma dessas funcionalidades é a visualização das vagas em um mapa interativo, onde o administrador do sistema pode ativar ou desativar as vagas adicionadas ao sistema.

Figura 8 – Mapa interativo das vagas



Fonte: feito pelo autor

6.2 CLIENTE

Foi desenvolvida uma aplicação para dispositivos móveis com sistema operacional Android. O sistema apresenta algumas funcionalidades para auxiliar os usuários do estacionamento, objetivo do presente trabalho.

A primeira funcionalidade a ser apresentada é a visualização das vagas do estacionamento em um mapa interativo, onde são exibidas as vagas disponíveis e as vagas ocupadas, atualizadas em tempo real.

Figura 9 – Visualizar vagas



Fonte: feito pelo autor

Com o mapa aberto, o usuário pode selecionar uma das vagas para fazer uso do sistema de rotas do estacionamento. Ao selecionar a vaga, o cliente se comunica com o servidor para

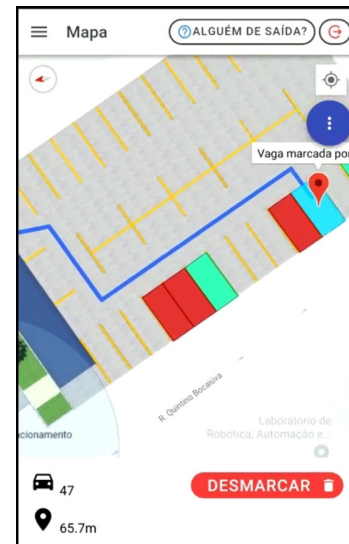
obter os dados referentes à rota a partir da entrada do estacionamento até a vaga selecionada, e a mesma será exibida no mapa. Além da rota, também é exibido o número da vaga e a distância até ela.

Figura 10 – Exibir rota - entrada



Fonte: feito pelo autor

Figura 11 – Exibir rota - vaga



Fonte: feito pelo autor

Seguindo com as funcionalidades implementadas, temos as opções de marcar e desmarcar uma vaga. Ao selecionar uma vaga que não esteja marcada, o usuário verá um botão com o texto "MARCAR ESTA VAGA". A marcação de vaga é útil para casos onde o usuário esquece onde estacionou. Ao abrir o mapa, o sistema exibirá a rota até a vaga marcada, a partir da entrada do estacionamento ou da saída do prédio - o sistema selecionará o ponto mais próximo da localização do usuário. Uma vaga marcada pode ser desmarcada através do botão "DESMARCAR".

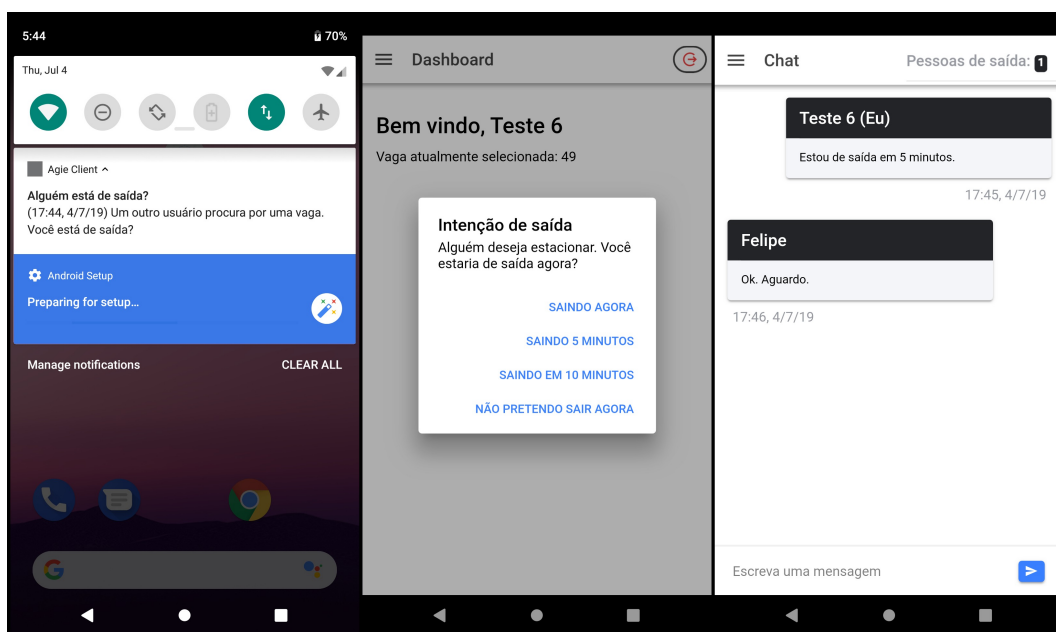
Além das funcionalidades citadas, foi implementada uma importante funcionalidade de comunicação entre os usuários. Caso o usuário não encontre vagas disponíveis, há a opção de enviar uma notificação aos demais usuários do sistema solicitando informações relativas ao horário de saída. Clicando no botão "ALGUÉM DE SAÍDA?", situado na barra superior do aplicativo, os usuários que possuem uma vaga marcada receberão uma notificação com opções para informar se pretendem liberar uma vaga nos próximos minutos. Quando um usuário responde à notificação, uma mensagem é enviada automaticamente em um chat, informando sobre a liberação de uma vaga nos minutos seguintes.

Figura 12 – Rota da vaga marcada



Fonte: feito pelo autor

Figura 13 – Verificação de intenção de saída



Fonte: feito pelo autor

7 CONCLUSÃO E TRABALHOS FUTUROS

O presente trabalho apresentou recursos para auxiliar o usuário na busca por vagas de estacionamento, tarefa muitas vezes estressante. Considerando os objetivos definidos, foi desenvolvida uma aplicação para dispositivos móveis a fim de auxiliar os usuários na busca por vagas de estacionamento.

Além da aplicação para dispositivos móveis, foi realizada uma evolução do sistema AGIE, desenvolvido por Lopes (2019), e que serviu como base para este trabalho. A implementação da funcionalidade de cadastro dos arquivos geométricos de áreas de estacionamento permite que o AGIE 2.0 seja escalável, auxiliando mais usuários no gerenciamento e monitoramento de vagas através dos mapas interativos.

Como objetivo para trabalhos futuros, seria de grande utilidade um sistema de navegação interna para complementar o sistema de rotas já implementado.

Além do sistema de navegação, também fica como objetivo para trabalhos futuros o desenvolvimento da aplicação para dispositivos móveis que utilizam o sistema operacional IOS, a fim de contemplar mais usuários.

REFERÊNCIAS

- AOKI, V. et al. Smart parking app-um aplicativo móvel para visualização de vagas em estacionamento inteligente. 2020. Citado na página 19.
- ARCGIS. *Shapefile file extensions*. 2011. Disponível em: <http://help.arcgis.com/en/arcgisdesktop/10.0/help/index.html#/Shapefile_file_extensions/005600000003000000/>. Acesso em: 16 jan, 2021. Citado na página 15.
- AZUMA, D. *RGeo*. 2012. Disponível em: <<https://www.w3.org/TR/geolocation-API/>>. Acesso em: 15 jan, 2021. Nenhuma citação no texto.
- BÄCHLE, M.; KIRCHBERG, P. Ruby on rails. *IEEE software*, IEEE, v. 24, n. 6, p. 105–108, 2007. Citado na página 13.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. *Software architecture in practice*. [S.l.]: Addison-Wesley Professional, 2012. Citado na página 12.
- CAI, W.; ZHANG, D.; PAN, Y. Implementation of smart parking guidance system based on parking lots sensors networks. In: IEEE. *2015 IEEE 16th International Conference on Communication Technology (ICCT)*. [S.l.], 2015. p. 419–424. Citado na página 18.
- CASANOVA, M. A. et al. *Banco de dados geográficos*. [S.l.]: MundoGEO Curitiba, 2005. Citado na página 13.
- DAVIS, C.; CÂMARA, G.; MONTEIRO, A. M. Introdução à ciência da geoinformação. *INPE, São José dos Campos: São Paulo*, v. 22, n. 03, p. 2010, 2001. Citado na página 13.
- DENATRAN. *Frota de veículos, por tipo e com placa, segundo as Grandes Regiões e Unidades da Federação - NOV / 2020*. 2020. Disponível em: <<https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-denatran/frota-de-veiculos-2020>>. Acesso em: 19 jan. 2021. Citado na página 10.
- DIJKSTRA, E. W. The humble programmer. *Commun. ACM*, v. 15, n. 10, p. 859–866, 1972. Citado na página 12.
- ESRI. Esri shapefile technical description. *Comput. Stat*, 1998. Citado na página 14.
- FREEMAN, A. *Pro Angular 6*. 3. ed. [S.l.]: Apress, 2018. ISBN 1484236483,9781484236482. Citado na página 13.
- GAMMA, E. *Design patterns: elements of reusable object-oriented software*. [S.l.]: Pearson Education India, 1995. Citado na página 13.
- GOOGLE. *Google Maps Platform*. 2019. Disponível em: <<https://cloud.google.com/maps-platform/maps/?hl=pt-br>>. Acesso em: 14 jun. 2019. Citado na página 17.
- HACKELOEER, A. et al. Georeferencing: a review of methods and applications. *Annals of GIS*, Taylor & Francis, v. 20, n. 1, p. 61–69, 2014. Citado na página 14.
- HASEGAWA, J. K. et al. Sistema de localização e navegação apoiado por gps. In: *Congresso Brasileiro de Cartografia*. [S.l.: s.n.], 1999. v. 19. Citado 2 vezes nas páginas 15 e 16.

- IBGE. *O que é um sistema geodésico de referência?* 2019. Disponível em: <<https://ww2.ibge.gov.br/home/geociencias/geodesia/pmrgr/faq.shtm>>. Acesso em: 03 jul. 2019. Citado na página 16.
- IBGE. *Projeção da população do Brasil e das Unidades da Federação*. 2021. Disponível em: <<https://www.ibge.gov.br/apps/populacao/projecao/>>. Acesso em: 19 jan. 2021. Citado na página 10.
- IBM. *Geographic coordinate system*. 2016. Disponível em: <https://www.ibm.com/support/knowledgecenter/en/SSGU8G_12.1.0/com.ibm.spatial.doc/ids_spat_407.htm>. Acesso em: 9 mai, 2019. Citado na página 16.
- IONIC. *Ionic - Cross-Platform Mobile App Development*. 2019. Disponível em: <ionicframework.com>. Acesso em: 24 abr, 2019. Citado 2 vezes nas páginas 13 e 21.
- KUMAR, M. World geodetic system 1984: A modern and accurate global reference frame. *Marine Geodesy*, Taylor & Francis, v. 12, n. 2, p. 117–126, 1988. Citado na página 16.
- LOPES, K. C. S. Agie - sistema para gestão de estacionamento. 2019. Citado 3 vezes nas páginas 18, 20 e 32.
- MASSE, M. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. [S.l.]: "O'Reilly Media, Inc.", 2011. Citado na página 15.
- NGA, N. G.-I. A. *WORLD GEODETIC SYSTEM 1984*. 2014. Disponível em: <http://earth-info.nga.mil/GandG/update/index.php?dir=wgs84&action=wgs84#tab_wgs84-res>. Acesso em: 9 mai, 2019. Citado na página 16.
- PADGITT, A. J.; HUND, A. M. How good are these directions? determining direction quality and wayfinding efficiency. *Journal of Environmental Psychology*, Elsevier, v. 32, n. 2, p. 164–172, 2012. Citado na página 11.
- ROSA, R.; BRITO, J. L. S. Introdução ao geoprocessamento. *Uberlândia: Universidades Federais de Uberlândia*, 1996. Citado 3 vezes nas páginas 13, 14 e 15.
- RUBY, S.; THOMAS, D. et al. *Agile web development with rails*. [S.l.]: Raleigh, NC: Pragmatic Bookshelf, 2009. Citado 2 vezes nas páginas 12 e 21.
- SOMMERVILLE, I. *Software engineering 9th ed.* [S.l.]: Pearson Education, Inc, 2011. Citado na página 12.
- W3C. *Geolocation API Specification 2nd Edition*. 2016. Disponível em: <<https://www.w3.org/TR/geolocation-API/>>. Acesso em: 3 mai, 2019. Citado na página 15.
- WADE, T.; SOMMER, S. et al. *A to Z GIS, An illustrated dictionary of geographic information systems*. [S.l.]: Esri Press, 2006. Citado na página 14.