



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

DAYVID EMERSON SILVA FERREIRA

**TOPIN: SISTEMA DE INFORMAÇÃO PARA O TRANSPORTE PÚBLICO DE PICOS**

PICOS, PIAUÍ

2018

DAYVID EMERSON SILVA FERREIRA

TOPIN: SISTEMA DE INFORMAÇÃO PARA O TRANSPORTE PÚBLICO DE PICOS

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

**Orientador:** Prof. Esp. Jesiel Viana da Silva

PICOS, PIAUÍ

2018

DAYVID EMERSON SILVA FERREIRA

TOPIN: SISTEMA DE INFORMAÇÃO PARA O TRANSPORTE PÚBLICO DE PICOS

Trabalho de conclusão de curso apresentado à Banca Examinadora como requisito parcial para aprovação na disciplina de Trabalho de Conclusão de Curso II 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 02 de Maio de 2018.

BANCA EXAMINADORA:

---

**Prof. Esp. Jesiel Viana da Silva**

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

---

**Prof. M<sup>e</sup>. Joao Paulo Lima do Nascimento**

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

---

**Prof. M<sup>e</sup>. Aislan Rafael Rodrigues de Sousa**

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

PICOS, PIAUÍ

2018

## RESUMO

A cidade de Picos é a terceira maior cidade do estado do Piauí e o transporte público da cidade se tornou um dos principais meios de locomoção para estudantes e trabalhadores na cidade. Apesar da utilização constante do transporte coletivo um problema enfrentado diariamente pelos passageiros é a dificuldade de acesso a informações básicas da prestação do serviço como as linhas e horários disponíveis. O que é um grande problema para passageiros que não utilizam o serviço com frequência ou até mesmo para os passageiros habituais quando necessitam utilizar um serviço em um horário que não está habituado.

Este trabalho visa o desenvolvimento de um sistema de informação com o objetivo de atender as necessidades dos passageiros, e para isso, o mesmo foi dividido em duas aplicações: sistema *web* para uso das empresas de transporte público e um aplicativo móvel para os passageiros, o conjunto dessas soluções foi nomeado de Topin. Este *software* que permite às empresas registrarem informações sobre os transportes públicos, como: linhas, horários, trajetos e pontos de referência, bem como possibilita que os passageiros consultem estas informações, além de outras funcionalidades que utilizam essas informações para melhorar a experiência dos passageiros no uso de transporte público na cidade de Picos.

**Palavras-chaves:** transporte público. passageiro. empresa de transporte público. sistema de informação, aplicativo móvel.

## **ABSTRACT**

The city of Picos is the third largest city in the state of Piauí and public transportation has become one of the main means of transportation for students and employees in the city. Despite the constant use of public transport, a problem faced by passengers on a daily basis is the difficulty of accessing basic information on the provision of the service, such as available lines and schedules. which is a major problem for passengers who do not use the service frequently or even for regular passengers when they need to use a service at a schedule they are not used to.

This work aims to develop an information system to meet the needs of passengers, and for this, it was divided into two applications: web system for use by public transport companies and a mobile application for passengers, the set of these solutions was named Topin. This software allows companies to record information on public transport, such as lines, timetables, routes and landmarks, as well as allowing passengers to consult this information, as well as other features that use this information to improve the use of public transport in the city of Picos.

**Key-words:** public transportation. passenger. public transport company. information system, mobile application.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Componentes De Um Sistema De Informação . . . . .	13
Figura 2 – Visão Geral da UML . . . . .	17
Figura 3 – Ciclo de Vida do Scrum . . . . .	21
Figura 4 – Exemplo de Kanban . . . . .	21
Figura 5 – Moovit - Tela de Visualização de Linhas . . . . .	23
Figura 6 – CittaMobi - Tela de Busca de Rotas . . . . .	24
Figura 7 – Trafi - Tela de Detalhes da Linha Seleccionada . . . . .	25
Figura 8 – Grau de Satisfação dos Usuários de Transporte Público de Picos . .	27
Figura 9 – Gráfico dos Problemas Levantados do Transporte Público de Picos .	28
Figura 10 – Gráfico Que Determina Onde os Passageiros Obtêm Informações .	29
Figura 11 – Gráfico Que Determina Onde os Passageiros Efetuam Reclamação	29
Figura 12 – Diagrama de Casos de Uso do Administrador . . . . .	35
Figura 13 – Diagrama de Casos de Uso do Passageiro . . . . .	36
Figura 14 – Diagrama Entidade-Relacionamento . . . . .	37
Figura 15 – Arquitetura Model-Template-View . . . . .	38
Figura 16 – Arquitetura Model-View-Presenter . . . . .	39
Figura 17 – Arquitetura REST . . . . .	40
Figura 18 – Arquitetura de Funcionamento do Django . . . . .	43
Figura 19 – Arquitetura de Funcionamento do GIT . . . . .	47
Figura 20 – Topin para Empresa - Tela de Cadastro da Linha . . . . .	48
Figura 21 – Topin para Empresa - Tela de Cadastro dos Horários e Trajeto da Linha	49
Figura 22 – Topin para Empresa - Tela de Cadastro dos Marcadores . . . . .	50
Figura 23 – Topin para Empresa - Tela de Visualização das Reclamações . . . .	51
Figura 24 – Topin para Passageiros - Tela de Visualização das Linhas Disponíveis	52
Figura 25 – Topin para Passageiros - Tela de Visualização da Linha Seleccionada	52
Figura 26 – Topin para Passageiros - Tela de Visualização dos Horários da Linha	53
Figura 27 – Topin para Passageiros - Tela de Visualização dos Pontos da Linha	54
Figura 28 – Topin para Passageiros - Tela de Registro de Feedback . . . . .	54

## LISTA DE TABELAS

Tabela 1 – Principais Métodos do Protocolo HTTP . . . . .	19
Tabela 2 – Comparativo das Soluções Semelhantes . . . . .	26
Tabela 3 – Requisitos Funcionais do Administrador da Empresa de Transporte Público . . . . .	31
Tabela 4 – Requisitos Funcionais do Passageiro . . . . .	32
Tabela 5 – Requisitos Não Funcionais do Administrador da Empresa de Transporte Público . . . . .	33
Tabela 6 – Requisitos Não Funcionais do Passageiro . . . . .	34
Tabela 7 – Recursos Disponíveis na API . . . . .	40
Tabela 8 – Tecnologias Utilizadas no Sistema Web . . . . .	41
Tabela 9 – Tecnologias Utilizadas no Aplicativo Móvel . . . . .	42

## LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BRT	Bus Rapid Transit
DER	Diagrama Entidade-Relacionamento
GIS	Geographic Information System
GPS	Global Positioning System
GUI	Graphical User Interface
ITS	Intelligent Transportation System
JSON	JavaScript Object Notation
MER	Modelo Entidade-Relacionamento
MTV	Model Template View
MTV	Model-Template-View
MVC	Model-View-Controller
MVP	Model-View-Presenter
ORM	Object/Relational Mapping
REST	Representational State Transfer
SGBD	Sistema de Gerenciamento de Banco de Dados
SIG	Sistema de Informação Gerencial
UI	User Interface
UML	Unified Modeling Language
UX	User Experience



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>10</b>
1.1	Motivação	10
1.2	Justificativa	10
1.3	Problematização	11
1.4	Objetivo Geral	11
1.5	Objetivos Específicos	12
1.6	Estrutura do Trabalho	12
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>13</b>
2.1	Sistema de Informação	13
2.1.1	Sistema Inteligente de Transporte	13
2.1.2	Sistema de Informação Geográfica	14
2.1.3	Sistema de Posicionamento Global	14
2.2	Engenharia de Software	15
2.2.1	Análise de Requisitos	15
2.2.1.1	Requisitos Funcionais	16
2.2.1.2	Requisitos Não Funcionais	16
2.2.2	Linguagem de Modelagem Unificada	16
2.2.2.1	Diagrama de Casos de Uso	17
2.2.2.2	Modelo Entidade-Relacionamento	17
2.2.3	Arquitetura de Software	18
2.2.4	Interface de Programação de Aplicativos	18
2.2.4.1	Transferência de Estado Representacional	19
2.2.4.2	Protocolo de Transferência de Hipertexto	19
2.2.5	Métodos Ágeis	19
2.2.5.1	Scrum	20
2.2.5.2	Kanban	21
<b>3</b>	<b>ESTADO DA ARTE</b>	<b>22</b>
3.1	Solução Atual	22
3.2	Soluções Semelhantes	22
3.2.1	Moovit	22
3.2.2	CittaMobi	23
3.2.3	Trafi	24
3.3	Análise das Soluções	25

<b>4</b>	<b>METODOLOGIA</b>	<b>27</b>
<b>5</b>	<b>TOPIN</b>	<b>31</b>
5.1	Análise de Requisitos	31
5.1.1	Requisitos Funcionais	31
5.1.2	Requisitos Não Funcionais	33
5.1.3	Modelagem do Sistema	34
5.1.3.1	Diagrama de Casos de Uso	35
5.1.3.2	Modelo Entidade-Relacionamento	37
5.2	Arquitetura do Software	37
5.2.1	Aplicativo Web	37
5.2.2	Aplicativo Móvel	38
5.2.3	API	39
5.3	Tecnologias Utilizadas	40
5.3.1	Python	42
5.3.2	Django	43
5.3.2.1	Django Rest Framework	43
5.3.2.2	GeoDjango	44
5.3.3	PostgreSQL	44
5.3.3.1	PostGIS	45
5.3.4	Java	45
5.3.5	SQLite	46
5.3.6	GIT	46
5.4	Funcionalidades	47
5.4.1	Administradores de Empresa de Transporte Público	48
5.4.2	Passageiro	51
<b>6</b>	<b>CONSIDERAÇÕES FINAIS</b>	<b>55</b>
6.1	Conclusão	55
6.2	Trabalhos Futuros	55
	<b>REFERÊNCIAS</b>	<b>57</b>

## 1 INTRODUÇÃO

Neste capítulo serão apresentados conceitos inerentes ao entendimento do trabalho, a motivação, a justificativa, o problema, os objetivos gerais e específicos, bem como a estrutura do trabalho.

### 1.1 MOTIVAÇÃO

Segundo Nascimento e Lima (2017) o transporte público urbano é um importante mecanismo para o deslocamento de pessoas dentro das cidades permitindo a mobilidade da população entre várias partes da cidade.

A aplicação de tecnologias da informação para solucionar problemas relacionados ao transporte público estão em constante expansão, conhecidas como Sistemas Inteligentes de Transporte Inteligente (em inglês *Intelligent Transportation System* - ITS). Segundo NTU (2013) os ITS consistem na aplicação de um conjunto de tecnologias em constante evolução a problemas comuns do transporte público, como a falta de informação e de planejamento, como também os congestionamentos, as contingências, etc.

No Brasil, os ITS estão em estágios iniciais se comparados com países desenvolvidos, porém têm mostrado desenvolvimentos importantes nos últimos anos, especialmente com as realizações de eventos como a Copa do Mundo em 2014 e os Jogos Olímpicos no Rio em 2016 (ALVARADO; JÚNIOR, 2016). Muitas das soluções desenvolvidas para auxiliar os turistas a se locomover na cidade, bem como acompanhar o andamento destes eventos, foram criadas para dispositivos móveis, devido ao alto índice de uso de *smartphones* no mundo.

Segundo dados da 28ª Pesquisa Anual do Uso de Tecnologia da Informação realizada pela Fundação Getúlio Vargas de São Paulo (FGV-SP) o Brasil terá um *smartphone* em uso por habitante até o final de 2017. A pesquisa nacional sobre os hábitos de utilização da *internet* no Brasil, feita pela Fecomércio-RJ e Instituto Ipsos, mostra que o *smartphone* se consolidou como principal meio para acessar a internet no país, utilizado por 69% dos internautas em 2016 (Agência Brasil, 2017).

### 1.2 JUSTIFICATIVA

Picos é a terceira maior cidade do estado do Piauí, com sua população estimada pelo Instituto Brasileiro de Geografia e Estatística (IBGE, 2010) de 73.414 habitantes, sendo que 79% da população vive na zona urbana e, com base em dados oferecidos

por uma empresa que presta serviço de transporte coletivo na cidade de Picos, são vendidas em média 71.854 passagens mensalmente.

Diversos órgãos gestores de transporte público têm aplicado inúmeros esforços para melhorar a qualidade dos serviços prestados (NTU, 2013). A modernização da forma que as informações são entregues aos passageiros são de suma importância para a garantia de uma boa experiência do usuário.

Após uma série de entrevistas e a aplicação de questionários com usuários de ônibus é notável que eles estão insatisfeitos com o serviço prestado na região de Picos, principalmente pela falta de conhecimento das linhas, rotas e horários que a empresa oferece e, com isso, se torna necessário o levantamento dos problemas acerca da insatisfação do passageiro, como também o desenvolvimento de uma solução que reduza significativamente os problemas encontrados.

### **1.3 PROBLEMATIZAÇÃO**

Na cidade de Picos existe uma defasagem na entrega de informações para os passageiros, atualmente não existe canal de comunicação para entrega de informações como rotas, linhas e horários do transporte para a cidade, o que leva os passageiros a aprenderem e compartilharem entre si os horários, o que se torna um grande problema ao usuário do transporte público quando precisar utilizar o transporte em um horário que não esteja acostumado e, por este motivo, faz com que ele fique esperando por mais tempo que o necessário na parada, já que não existe se quer uma folha com os horários nas paradas.

Portanto é perceptível a necessidade de um maior controle sobre as informações e principalmente um meio para compartilhar essas informações com a população com o objetivo de reduzir o tempo de espera em paradas, bem como melhorar a experiência que o passageiro tem para com o uso de transporte público na cidade.

### **1.4 OBJETIVO GERAL**

O objetivo geral deste trabalho de conclusão de curso é coletar dados que possibilitem a criação de um sistema de informação aos usuários de ônibus da cidade de Picos, bem como fornecer um meio para que as empresas prestadoras deste serviço gerenciem os dados informados aos passageiros.

A proposta está focada no problema da insatisfação do passageiro em relação ao uso do transporte público, que se deve a dificuldade que os passageiros possuem em acessar as informações referentes ao transporte público. Para isso, propõe-se o desenvolvimento de uma aplicação para dispositivos móveis que possibilite ao seus usuários visualizar informações que o auxiliem na utilização do transporte público,

como também uma plataforma *web* que permita as empresas registrarem os dados do serviço.

## 1.5 OBJETIVOS ESPECÍFICOS

Para a obtenção do objetivo geral, fez-se necessário a sua divisão em algumas etapas distintas, são estas:

- Identificação dos problemas dos usuários de ônibus da cidade de Picos;
- Elicitação das necessidades dos usuários que possam ser automatizadas por meio de um sistema computacional;
- Elaborar uma abordagem que possa reduzir a maior quantidade possível dos problemas encontrados, bem como atenda as necessidades elicítadas;
- Desenvolver um sistema que atenda a abordagem citada acima;
- Prover uma ferramenta para gerenciar os dados referentes ao serviço de transporte público;
- Prover uma ferramenta móvel de acesso aos dados referentes ao uso do transporte público;

## 1.6 ESTRUTURA DO TRABALHO

Este trabalho está separado em seis capítulos: Introdução, Fundamentação Teórica, Estado da Arte, Metodologia, Topin e Considerações Finais, respectivamente, em que o primeiro capítulo trata-se de definir a motivação, a justificativa, o problema e os objetivos deste trabalho. No segundo capítulo aborda os conceitos necessários para a compreensão e implementação deste trabalho. No terceiro capítulo são apresentadas algumas soluções disponíveis para o problema proposto, bem como uma comparação das funcionalidades disponíveis entre as soluções encontradas e o sistema implementado no presente trabalho. No quarto capítulo é descrito o método de implementação e pesquisa utilizado para o desenvolvimento do trabalho em questão. No quinto capítulo é realizada uma descrição das funcionalidades, como se deve o seu funcionamento, além de mostrar algumas das principais tecnologias utilizadas para o desenvolvimento da solução proposta. O sexto e último capítulo são apresentadas as considerações finais e trabalhos futuros.

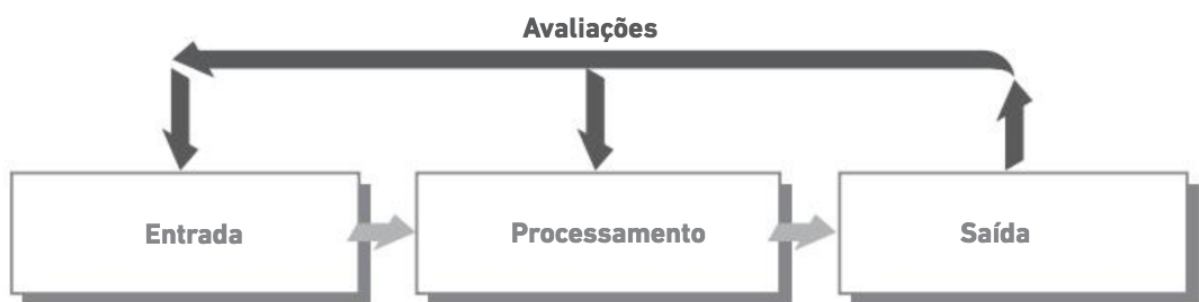
## 2 FUNDAMENTAÇÃO TEÓRICA

A seguir são apresentados conceitos, visões e abordagens de autores quanto aos temas necessários para o entendimento desse trabalho, bem como a descrição das tecnologias utilizadas para o desenvolvimento da solução Topin.

### 2.1 SISTEMA DE INFORMAÇÃO

Para Stair e Reynolds (2011) um Sistema de Informação é composto por um conjunto de componentes (Figura 1) inter-relacionados com o intuito de coletar, manipular, armazenar e disseminar dados e informações, além de fornecer um mecanismo de realimentação para atingir um objetivo.

Figura 1 – Componentes De Um Sistema De Informação



Fonte: Cengage Learning

#### 2.1.1 Sistema Inteligente de Transporte

Segundo Figueiredo et al. (2005) os Sistemas Inteligentes de Transporte são uma tendência mundial em constante crescimento que entrou em evidência após sociedade moderna dar mais importância a mobilidade e, com isso, os ITS passaram a contemplar os setores públicos e privados colocando em destaque assuntos de índole política, econômica e social.

Ainda sobre a visão de Figueiredo et al. (2005) os ITS pretendem conciliar a cultura, a indústria, a econômica, a natureza, o ambiente e por fim o estilo e qualidade de vida do ser humano. A definição dos ITS não apenas revoluciona a interação entre pessoas e veículos, como também possibilita o desenvolvimento de novas relações espaciais e, por este motivo, viabiliza a comunicação entre veículos e pessoas, constituindo uma sociedade avançada baseada em informação.

De acordo com a ISO/TR 10992 (2012) os ITS se dividem nas seguintes categorias:

- Gerenciamento de tráfego;
- Informações aos usuários;
- Gerenciamento das rodovias;
- Sistemas de condução assistida de última geração;
- Gerenciamento de veículos comerciais;
- Gerenciamento do transporte público;
- Pagamento eletrônico;
- Respostas a incidentes e desastres.

Das oito categorias abordadas pelos ITS, os sistemas de informações aos usuários vem sendo desenvolvido em várias cidades brasileiras (REVISTA OPARA, 2014). Nesta categoria encontram-se aplicações que disponibilizam informações antes do deslocamento, informações durante o deslocamento e serviços pessoais de informação para os usuários.

### **2.1.2 Sistema de Informação Geográfica**

Os Sistemas de Informação Geográfica (em inglês *Geographic Information System* - GIS) se tornaram um importante instrumento para a resolução de problemas relacionados ao transporte. Inúmeros estudos se utilizam de GIS no planejamento, gestão, operação e análise de sistemas de transporte, as aplicações dos Sistemas de Informação Geográfica nos transportes são diversificadas, entre elas pode-se citar o transporte coletivo urbano, rodoviário, de carga e engenharia de tráfego. (DANTAS; TACO; YAMASHITA, 1996)

Para Rose (2001) os Sistemas de Informação Geográfica são a melhor ferramenta para solucionar problemas de organização de dados em modelos espaciais. Inúmeras empresas privadas e órgãos governamentais tomam decisões relacionadas ao planejamento com base em um GIS, utilizando de seus recursos em relação a ferramentas de gerenciamento, bancos de dados e processamento de dados. os Sistemas de Informação Geográfica também tem sido elemento chave para aprimorar o gerenciamento de transportes existentes.

### **2.1.3 Sistema de Posicionamento Global**

Para Vázquez (1999) os Sistemas de Posicionamento Global (em inglês *Global Positioning System* - GPS) funcionam de modo a permitir que se determine posições precisas para cada ponto ao redor do globo terrestre, facilitando assim a possibilidade de inserir dados associados a um ponto no espaço, como por exemplo, determinar a

localização exata de um hotel ou, inclusive, a localização em tempo real de um objeto em movimento.

## 2.2 ENGENHARIA DE SOFTWARE

Para Sommerville (2011) a Engenharia de Software é uma disciplina da engenharia que aborda todos os aspectos da produção de um *software* desde os estágios iniciais da especificação do sistema até a sua manutenção, apesar de não se restringir as atividades de implementação, dado que também se preocupa com o gerenciamento de projetos e a criação de novas ferramentas, métodos e teorias quando não houver alternativa viável, para suportar a produção do *software*, entretanto existem 4 atividades fundamentais para a Engenharia de Software:

- Especificação do *software*;
- Desenvolvimento de *software*;
- Validação de *software*;
- Evolução do *software*.

Segundo Rezende (2005) a Engenharia de Software é um conjunto de métodos para implementação e manutenção de aplicações segmentadas, que estejam de acordo com as características abaixo:

- Processo dinâmico, integrado e inteligente de soluções tecnológicas;
- Adequação aos requisitos funcionais do negócio do cliente e seus respectivos procedimentos pertinentes;
- Efetivação de padrões de qualidade, produtividade e efetividade em suas atividades e produtos;
- Fundamentação na Tecnologia da Informação disponível, viável, oportuna e personalizada;
- Planejamento e gestão de atividades, recursos, custos e datas.

### 2.2.1 Análise de Requisitos

Para a Engenharia de Software a documentação e levantamento dos requisitos é um dos problemas básicos encontrados durante o planejamento do desenvolvimento de um produto de *software*, porque se o levantamento é feito da forma correta, os requisitos implícitos, que são os requisitos cobrados por clientes que não foram informados no levantamento de requisitos, são minimizado de modo a evitar problemas no andamento da implementação do sistema, pois requisitos não documentados dificilmente entram



no desenho do produto ocasionando em alterações não desejadas e inesperadas, bem como a insatisfação do cliente. (PAULA FILHO, 2003)

Os requisitos de *software* são as funções ou características pertencentes a um *software*, requisitadas pelo cliente ou usuário para resolver um problema ou atingir um objetivo. (IEEE, 1990)

O valor de um produto de *software* é proveniente de suas características ou requisitos. Ao se trabalhar com desenvolvimento de sistemas esses requisitos são divididos em duas categorias: requisitos funcionais e requisitos não funcionais. (PAULA FILHO, 2003)

### 2.2.1.1 Requisitos Funcionais

Os requisitos funcionais podem ser definidos como as funções ou atividades que são ou viram a ser desempenhadas por um sistema computacional. Estes devem ser definidos de forma clara e posteriormente relatados explicitamente. (SOMMERVILLE, 2011)

### 2.2.1.2 Requisitos Não Funcionais

Para Cysneiros e Leite (1997) os requisitos não funcionais, diferentemente dos requisitos funcionais, não estão relacionados a uma função que o sistema deve desempenhar, mas sim aos comportamentos e restrições que este *software* deve satisfazer.

## 2.2.2 Linguagem de Modelagem Unificada

Linguagem de Modelagem Unificada (em inglês *Unified Modeling Language* - UML) é uma linguagem gráfica para elaboração de artefatos que irão permitir a visualização, especificação, construção e documentação de artefatos de sistemas complexos de *software*. A UML é um forma de garantir a padronização no momento de preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais tais como processos de negócios e funções do sistema, além de itens concretos como as classes escritas em determinada linguagem de programação, esquemas de bancos de dados e componentes de *software* reutilizáveis. (BOOCH; RUMBAUGH; JACOBSON, 2006)

Como pode ser visto na Figura 2, a UML é formada por um conjunto de diagramas que podem ser classificados em três categorias:

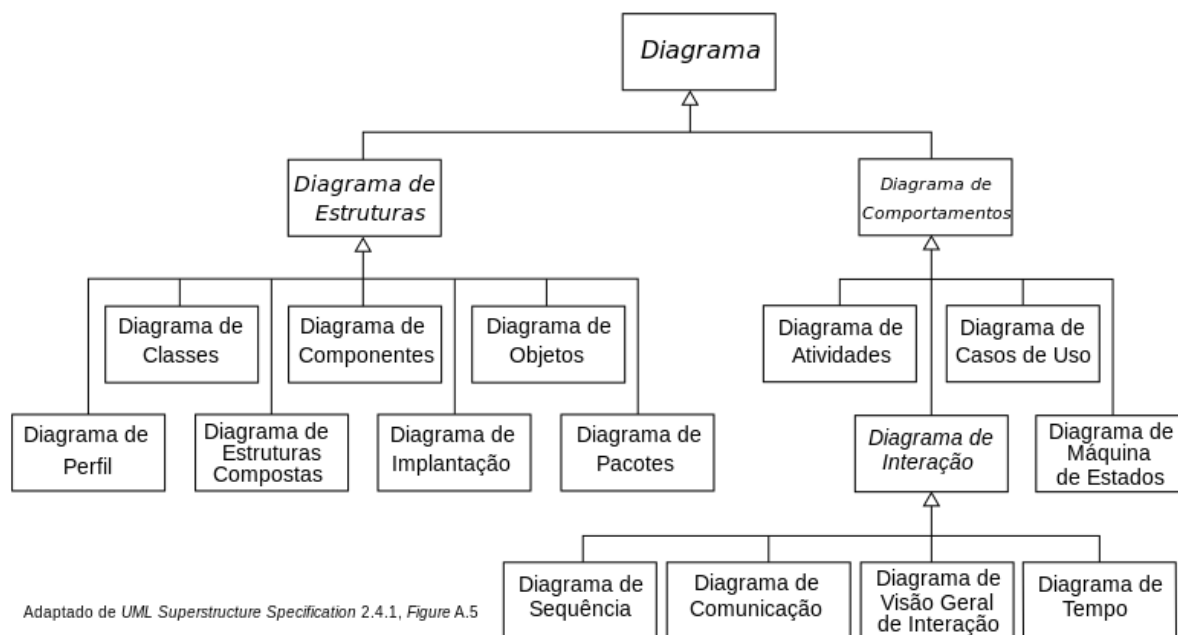
- **Diagramas de Estruturas:** Descreve o sistema de forma estática, abordando as partes referentes a objetos, classes, atributos, operações, bem como os relacionamentos entre si.

- **Diagramas de Comportamento:** Descreve o funcionamento dos módulos e os processos do negócio que possuem relação com o sistema.
- **Diagramas de Interação:** Subgrupo dos diagramas de comportamento responsável pela descrição das interações entre os objetos.

### 2.2.2.1 Diagrama de Casos de Uso

De acordo com Guedes (2018) o diagrama de casos de uso tem por definição o objetivo de permitir que qualquer pessoa com o mínimo de conhecimento sobre o problema possa compreender o comportamento externo do sistema pela perspectiva dos usuários, ou seja, apresenta uma visão geral das funcionalidades que o sistema deve oferecer aos usuários, sem considerar os detalhes de implementação.

Figura 2 – Visão Geral da UML



Fonte: Wikimedia Commons

### 2.2.2.2 Modelo Entidade-Relacionamento

O modelo entidade-relacionamento (MER) é representado visualmente por meio de um diagrama entidade-relacionamento (DER). O MER é considerado um padrão para a modelagem conceitual de bancos de dados desde sua criação em 1976, por Peter Chen.(HEUSER, 2009)

O MER tem o objetivo de representar de forma abstrata um banco de dados, interessam-se apenas os objetos sobre os quais existe a necessidade de armazenar as informações. Com ele é possível representar as entidades, relacionamentos e

cardinalidades de todos os objetos que compõem o *software* objeto deste trabalho e, a partir dele, é possível implementar o banco de dados que será utilizado neste projeto.(HEUSER, 2009)

### 2.2.3 Arquitetura de Software

De acordo com Bass, Clements e Kazman (2003) a arquitetura de *software* de um sistema computacional é formado por uma ou mais estruturas que compreendem todos os elementos do *software*, bem como suas propriedades visíveis externamente e o modo como estão interligadas.

Para Bass, Clements e Kazman (2003) a importância da arquitetura de *software* para o sistema se dá por meio de 3 tópicos principais, são estes:

- **Comunicação entre as partes interessadas.** A arquitetura de *software* representa uma abstração comum de um sistema que a maioria das partes interessadas no sistema podem usar como base para entendimento, negociação, consenso e comunicação.
- **Decisões antecipadas de design.** A arquitetura de *software* manifesta as primeiras decisões de projeto sobre um sistema, e essas ligações iniciais carregam um peso muito grande em relação ao desenvolvimento restante do sistema, sua implantação e sua vida de manutenção. Como também é a primeira oportunidade para avaliar o que será desenvolvido e analisar o melhor caminho a ser desempenhado pela equipe.
- **Modelo genérico.** A arquitetura de *software* constitui um modelo relativamente pequeno, genérico e de fácil compreensão que representa como um sistema é estruturado e como seus elementos trabalham em conjunto, bem como pode ser reproduzido e aplicado a outros sistemas.

Para Pressman (2005) arquitetura não é o *software* operacional. Pelo contrário, é uma representação que permite a um engenheiro de *software*:

- Analisar a eficiência do projeto em atender aos requisitos levantados.
- Considerar as alternativas de arquitetura em um estágio em que fazer mudanças no projeto ainda é processo relativamente fácil e menos custoso.
- Reduzir os riscos associados ao desenvolvimento do *software*.

### 2.2.4 Interface de Programação de Aplicativos

Masse (2011) afirma de forma sucinta que uma Interface de Programação de Aplicativos (em inglês *Application Programming Interface - API*) é um modo de expor um

conjunto de dados e funções para facilitar as interações entre programas de computador e permitir que troquem informações.

#### 2.2.4.1 Transferência de Estado Representacional

Fielding e Taylor (2002) definiram os princípios da arquitetura moderna da *internet* desenvolvendo um modelo de transferência de dados na *web*, nomeado de Transferência de Estado Representacional (em inglês *Representational State Transfer - REST*), este modelo consiste em um conjunto coordenado de restrições arquiteturais que tenta minimizar a latência e a comunicação de rede, maximizando ao mesmo tempo a independência e a escalabilidade da implementação dos componentes.

#### 2.2.4.2 Protocolo de Transferência de Hipertexto

Para Fielding et al. (1999) O Protocolo de Transferência de Hipertexto (em inglês *Hypertext Transfer Protocol - HTTP*) é um protocolo em nível de aplicativo para sistemas de informações hipermídia distribuídos e colaborativos. É um protocolo genérico, sem estado, que pode ser usado para muitas tarefas além de seu uso para hipertexto, como servidores de nomes e sistemas de gerenciamento de objetos distribuídos, através da extensão de seus métodos de solicitação, códigos de erro e cabeçalhos.

Tabela 1 – Principais Métodos do Protocolo HTTP

Método	Função
GET	Recuperar os dados de um recurso.
POST	Adicionar um novo recurso.
PUT	Atualizar os dados de um recurso.
PATCH	Atualizar apenas alguns dados de um recurso.
DELETE	Apagar um recurso.

Fonte: Autor

#### 2.2.5 Métodos Ágeis

Os métodos ágeis são métodos de desenvolvimento incremental baseados na entrega de pequenos incrementos, onde novas versões do sistema são implementadas e fornecidas aos clientes em prazos curtos, geralmente, de duas a três semanas. Eles colocam a participação do cliente como essencial no processo de desenvolvimento do *software* no intuito de obter um *feedback* rápido sobre as mudanças e requisitos do

sistema. Assim, podem minimizar a documentação utilizando-se de uma comunicação informal, ao invés de reuniões formais com documentos escritos. (SOMMERVILLE, 2011)

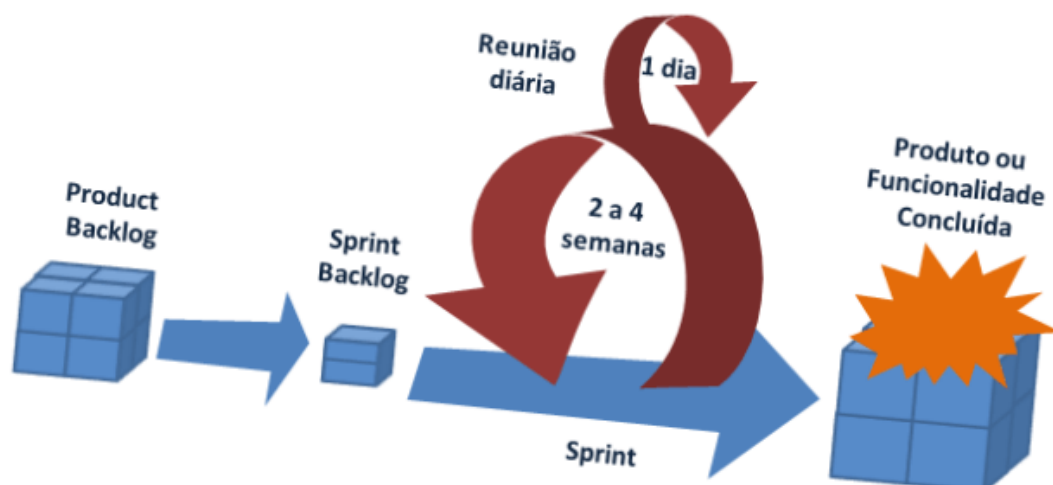
A filosofia dos métodos ágeis é refletida através do manifesto ágil que foi criado por grande parte dos principais desenvolvedores desses métodos, esse manifesto afirma que um método ágil deve valorizar os seguintes aspectos:

- Valorizar os indivíduos e as suas interações acima dos processos e ferramentas;
- Valorizar mais o *software* em funcionamento do que uma documentação abrangente;
- Valorizar a colaboração do cliente em relação à negociação de contratos;
- Valorizar a capacidade de responder a mudanças ao invés de seguir um plano.

#### 2.2.5.1 Scrum

O Scrum aborda todas as suas práticas em uma estrutura de processo incremental e iterativo, como pode ser visto na Figura 3 o ciclo maior representa a *Sprint Retrospective* em que demonstra uma iteração das atividades de desenvolvimento que ocorrem uma após a outra em pequenos intervalos de tempo, geralmente entre duas e quatro semanas. A saída de cada iteração consiste em um incremento do produto. O círculo menor representa as reuniões diárias (em inglês *Daily Scrum*) que ocorrem todos os dias durante a iteração, onde todos os membros da equipe se reúnem para discutir as atividades em desenvolvimento e efetuar as modificações apropriadas, se necessário. O ciclo de iterações deve ser repetido até que o objetivo do projeto tenha sido alcançado ou não esteja mais sendo financiado. (SCHWABER, 2004)

Figura 3 – Ciclo de Vida do Scrum

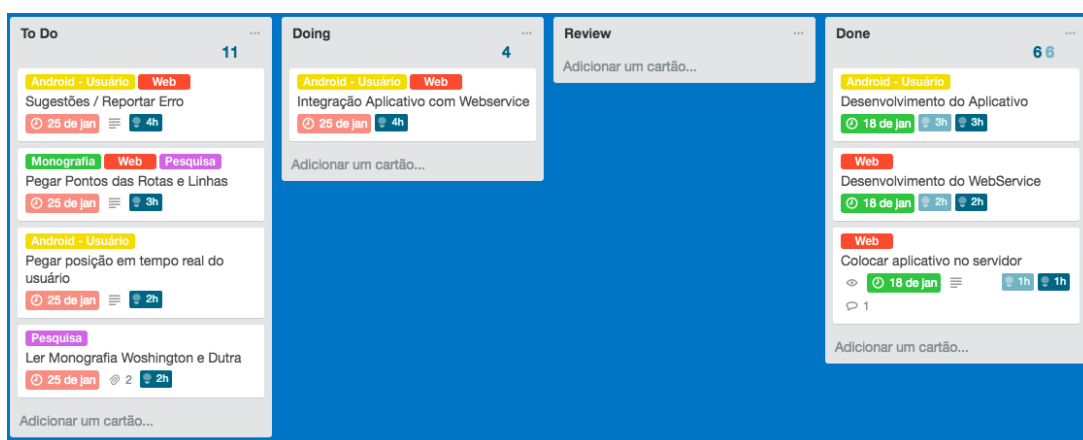


Fonte: Mind Master

### 2.2.5.2 Kanban

Segundo os autores Kniberg e Skarin (2010) o Kanban é aplicado de um modo em que as atividades que estão em processo de desenvolvimento devem ser limitadas e novas atividades só podem ser desempenhadas quando uma das atividades que estão em andamento são entregues ou movidas para uma etapa diferente do processo. O Kanban determina que um sinal visual deve ser produzido na tentativa de indicar que um novo trabalho pode ou não ser colocado em desenvolvimento e, assim, determinar se o trabalho atual estará em conformidade com o limite de acordado.

Figura 4 – Exemplo de Kanban



Fonte: Autor

### 3 ESTADO DA ARTE

A seguir é demonstrado a forma com que os usuários de ônibus resolvem parcialmente os problemas encontrados, além de uma análise detalhada a respeito de tecnologias utilizadas para resolver problemas semelhantes, como também, este capítulo tem a finalidade de evidenciar os principais critérios utilizados para implementar a solução.

#### 3.1 SOLUÇÃO ATUAL

Atualmente, por não existir um canal de comunicação apropriado para a obtenção de informações sobre o transporte público coletivo, os usuários de ônibus da cidade de Picos se comunicam com outros usuários de ônibus ou funcionários, como motoristas e cobradores, para obter as informações que necessitam para utilizar o transporte público. Contudo não é um meio eficiente de obtenção de informações visto que as pessoas que estão disponíveis para questionamento podem não ter todas as informações que o usuário deseja por se tratar de um dia ou horário específico.

#### 3.2 SOLUÇÕES SEMELHANTES

Existem aplicativos no mercado com o intuito de atender o mesmo problema abordado neste trabalho, portanto esta seção possui o objetivo de abordar as principais funcionalidades de três das principais aplicações encontradas e selecionadas pelo fato de cada uma possuir mais de 1 milhão de usuários e com uma avaliação positiva dos usuários, recebendo notas acima de 4 em uma escala de 1 a 5.

##### 3.2.1 Moovit

O Moovit <sup>1</sup> possibilita que os seus usuários acompanhem todo o trajeto pelo aplicativo, a partir do momento em que você escolhe o destino, até a sua chegada. Com ele o usuário consegue encontrar a melhor linha para chegar ao seu destino e recebe instruções sobre como chegar à estação ou parada mais próxima, do mesmo modo informa ao usuário quando está ele está chegando e em qual local descer. Como também, fornece programações atualizadas dos horários e atualizações de chegada em tempo real, recebidas do GPS dos ônibus e trens.

Com este aplicativo também é possível acompanhar em tempo real as informações atualizadas sobre atrasos e interrupções do serviço, engarrafamentos e novas obras, para lhe ajudar a navegar até o seu destino e, com isso, efetuar o planejamento

---

<sup>1</sup> <https://moovitapp.com/>

de viagens por meio da visualização de rotas que lhe ajudam a chegar ao seu destino, através da comparação dos horários de partida, tempo total de caminhada até a sua estação, e rotas combinando diferentes meios de transporte.

Também é possível que os usuários reportem problemas encontrados em estações, linhas e horários, para informar os passageiros nas proximidades o que está acontecendo.

Figura 5 – Moovit - Tela de Visualização de Linhas



Fonte: Autor

### 3.2.2 CittaMobi

O CittaMobi <sup>2</sup> é um aplicativo de acompanhamento dos transportes públicos que possibilita a previsão de tempo de chegada dos ônibus em tempo real, mostra a melhor rota de sua localização atual até o seu destino, vende bilhetes eletrônicos que funcionam como passagens para uso do ônibus, disponibiliza um botão para aviso de incidente grave e mostra o itinerários das linhas de ônibus.

Entre as funcionalidades do CittaMobi está a previsão de chegada dos ônibus que se dá por meio da coleta da posição dos ônibus em tempo real por meio de um GPS instalado no mesmo. Com o objetivo de garantir a precisão do horário de chegada em todas as paradas em seu percurso e, com isso, permite que os usuários percam menos tempo esperando.

<sup>2</sup> <https://www.cittamobi.com.br/home/>



Com a utilização deste aplicativo é possível encontrar o melhor caminho até o seu destino, incluindo trechos a pé, de trem, ônibus ou metrô. Todas estas possibilidades são levadas em consideração ao traçar a melhor rota e demonstrar em um mapa detalhado o percurso definido.

O aplicativo ainda permite que os usuários gravem seus pontos favoritos para facilitar o uso do aplicativo, pois assim é possível facilitar a busca de informações sobre o ponto favorito, como por exemplo o horário de chegada dos ônibus e as linhas que passam por ele.

Figura 6 – CittaMobi - Tela de Busca de Rotas



---

Fonte: Autor

### 3.2.3 Trafi

O Trafi <sup>3</sup> é um aplicativo de acompanhamento dos transportes públicos que possibilita o acompanhamento de sua cidade definida como favorita e a visualização dos detalhes em tempo real, todavia esta solução permanece funcional sem acesso a *internet*, limitando as suas funcionalidades, porém ainda é possível consultar informações importantes sobre as linhas, horários e exibir as rotas, em um mapa, das linhas disponíveis.

Com a utilização deste aplicativo é possível encontrar o melhor caminho até

---

<sup>3</sup> <https://www.trafi.com/>

o seu destino, conectando todas as suas opções de transporte que incluem trechos de trem, ônibus, metrô, bicicletas públicas, carros compartilhados e motoristas de aplicativos. Ao conectar todos os transportes disponíveis o Trafi mostra ao usuário sua melhor opção pra chegada ao seu destino com base nos transportes disponíveis para a região solicitada, oferecendo a possibilidade do usuário comparar e escolher a melhor rota com base na eficiência, tempo ou custo.

O aplicativo ainda permite que os usuários gravem seus seus destinos favoritos para facilitar o uso do aplicativo, pois assim é possível facilitar a busca de rotas disponíveis para o seu destino favorito com base na sua localização atual e, com isso, o usuário não precisa digitar o mesmo endereço de destino repetidas vezes.

Figura 7 – Trafi - Tela de Detalhes da Linha Seleccionada



Fonte: Autor

### 3.3 ANÁLISE DAS SOLUÇÕES

Ao analisar as soluções descritas neste capítulo, foram escolhidas algumas características que são essenciais para melhor atender a cidade de Picos. Todas as características escolhidas foram relacionadas as funcionalidades existentes em cada solução, As características selecionadas e a solução que as contempla podem ser vistas na Tabela 2.

Ao analisar a Tabela 2, é possível verificar que as soluções propostas neste

capítulo abordam 5 das 7 características escolhidas, contudo nosso objetivo principal se deve a insatisfação do uso de transporte público na região de Picos e nenhuma das soluções encontradas atende essa necessidade e, ao analisar as aplicações existentes é evidente que não é de interesse destes aplicativos atender cidades menores, focando seus esforços em grandes centros urbanos, com múltiplos recursos e serviços que não se fazem necessários em cidades menores.

Tabela 2 – Comparativo das Soluções Semelhantes

<b>FUNCIONALIDADE</b>	<b>MOOVIT</b>	<b>CITTAMOB</b>	<b>TRAFI</b>	<b>TOPIN</b>
Busca de Linhas	X	X	X	X
Visualiza os Horários das Linhas	X	X	X	X
Verifica Paradas Mais Próximas	X	X	X	X
Acompanha Transporte em Tempo Real	X	X	X	X
Busca de Linhas e Horários Offline	X		X	X
Atende a Cidade de Picos				X
Ambiente Administrativo Para Empresa de Transporte				X

Fonte: Autor

As características selecionadas para a comparação, foram determinadas como parâmetro para análise, com base em uma pesquisa efetuada com usuários do transporte público na região de Picos, que será melhor evidenciada no capítulo que se segue, portanto esses aplicativos podem ter mais funcionalidades que o Topin, todavia não possuem relação direta com o objetivo deste trabalho.

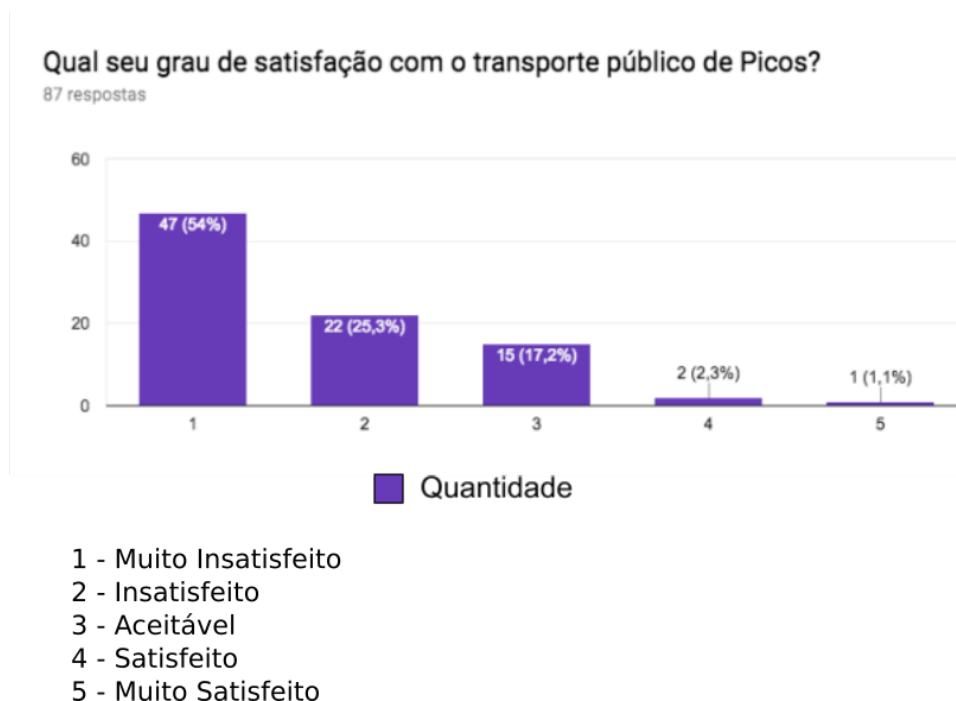
#### 4 METODOLOGIA

Este trabalho tem como seu principal objetivo desenvolver uma solução computacional que possibilite a melhoria significativa da experiência dos usuários de transporte público da cidade de Picos. Para chegar ao desenvolvimento, o desenvolvimento do projeto foi dividido em três etapas, começando com uma pesquisa objetiva com os passageiros, com o propósito de entender a realidade do problema que torna a utilização do transporte público insatisfatório.

Em uma pesquisa que contou com a participação de 87 pessoas que utilizam o transporte público em Picos, foi levantado os seguintes questionamentos:

- Qual seu grau de satisfação com o transporte público de Picos?
- Quando insatisfeito para quem você acha que pode e deve reclamar?
- Quais os principais problemas enfrentados ao utilizar ônibus em Picos?
- Qual sistema operacional do seu *smartphone*?

Figura 8 – Grau de Satisfação dos Usuários de Transporte Público de Picos



Fonte: Autor

O resultado obtido sobre a satisfação dos usuários pode ser visto na Figura 8, onde foi determinado que escolhessem um valor pertencente ao intervalo de 1 a 5 para medir o grau de satisfação do usuário, onde o menor número representava o

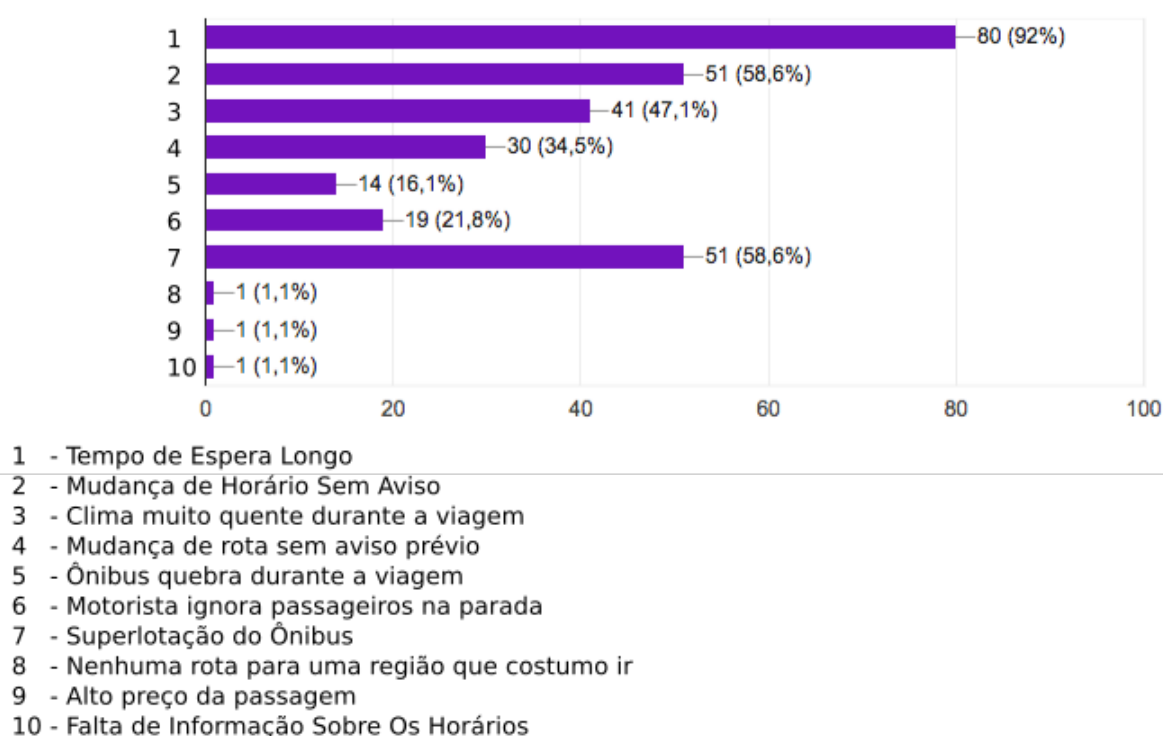
menor grau de satisfação possível, em que foi possível identificar que mais de 78% dos usuários estão insatisfeitos com o transporte público, contudo também foi percebido que a maioria dos usuários não possuem uma forma de obter informações confiáveis e não tem conhecimento de onde podem fazer reclamações.

Ainda nessa mesma pesquisa foi possível identificar que os principais problemas encontrados foram o tempo de espera e as mudanças de horários dos ônibus sem aviso prévio da empresa prestadora do serviço.

Figura 9 – Gráfico dos Problemas Levantados do Transporte Público de Picos

### Quais as principais problemas enfrentados ao utilizar ônibus em Picos? (Deixe em branco se não houver problemas)

87 respostas



Fonte: Autor

Na segunda etapa, mediante uma análise sobre os resultados da pesquisa com os passageiros e entrevistas realizadas com os fiscais e gestores do transporte público local, é nítido que a causa dos principais problemas é o fato dos usuários não terem acesso a informações sobre o funcionamento dos ônibus coletivos, tais como: horários, rotas e incidentes diários. Diante disso, constatou-se que para reduzir esses problemas é necessário desenvolver um meio de comunicação que forneça informações aos passageiros.

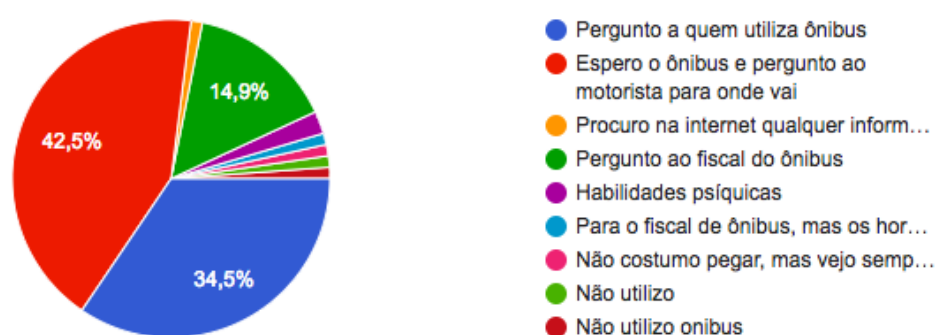
Ainda na segunda etapa é necessário que após a identificação e análise dos

problemas seja determinado os requisitos para o desenvolvimento de um sistema computacional que auxilie na redução dos problemas identificados, entretanto, foi determinado previamente que a solução desenvolvida envolvesse o compartilhamento de dados em tempo real, como também a consulta dessas informações por meio de um aplicativo móvel.

Figura 10 – Gráfico Que Determina Onde os Passageiros Obtêm Informações

### Como você consegue informações sobre os horários e rotas dos ônibus?

87 respostas

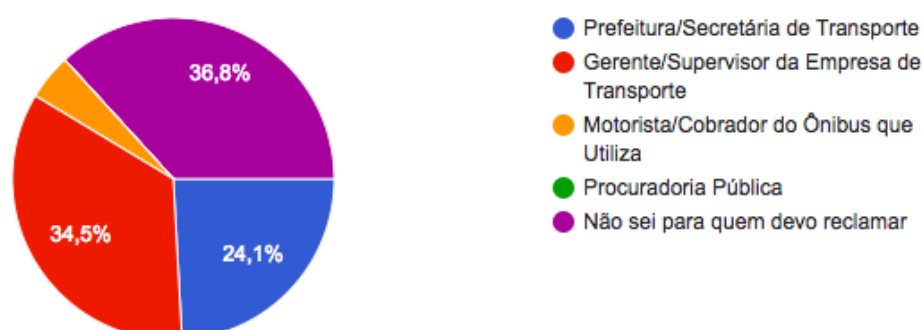


Fonte: Autor

Figura 11 – Gráfico Que Determina Onde os Passageiros Efetuam Reclamação

### Quando insatisfeito para quem você acha que pode e deve reclamar?

87 respostas



Fonte: Autor

E na terceira e última etapa será aplicado uma metodologia de desenvolvimento ágil para a implementação dos softwares necessários para o funcionamento dos re-

quisitos levantados na etapa 2, tal que, esse sistema será composto por um módulo de manutenção das informações providas ao usuário que será administrado pela empresa prestadora do serviço de transporte público e um módulo para consulta dessas informações pelos passageiros, como também serão implementadas funcionalidades que visam melhorar a comunicação direta entre os passageiros e os responsáveis pela prestação do serviço, por meio de formulários que permitam o passageiro efetuar reclamações e sugestões com a garantia de que os responsáveis recebam a mensagem, como também, permitir que as empresas enviem notificações para que todos os usuários tenham acesso rápido a informações que afetam diretamente a sua experiência de uso do transporte público urbano.

## 5 TOPIN

Durante o processo de desenvolvimento de um projeto, inúmeras ferramentas são utilizadas para auxiliar os membros da equipe de desenvolvimento a minimizar os possíveis problemas que a aplicação venha a possuir. Este capítulo tem o intuito de descrever as tecnologias utilizadas para a construção do projeto, bem como demonstrar as características da solução desenvolvida.

### 5.1 ANÁLISE DE REQUISITOS

#### 5.1.1 Requisitos Funcionais

Nesta seção serão descritas as principais funcionalidades referentes ao sistema de informação Topin, contudo os requisitos serão separados de forma a melhorar o entendimento sobre a aplicação como um todo, dividindo os requisitos referentes ao ambiente de gerenciamento das informações por parte da empresa prestadora de serviço de transporte público e a aplicação móvel disponibilizada para os passageiros.

#### Empresa de Transporte Público

Tabela 3 – Requisitos Funcionais do Administrador da Empresa de Transporte Público

ID	História de Usuário
RF001	Como administrador da empresa preciso registrar as linhas disponíveis, atribuindo as coordenadas geográficas que determinam o caminho percorrido de cada linha.
RF002	Como administrador da empresa preciso registrar as paradas de ônibus referenciadas pela sua geolocalização.
RF003	Como administrador da empresa preciso registrar os horários de partidas de cada linha e o dia da semana que aquele horário está disponível.
RF004	Como administrador da empresa preciso registrar marcadores de interesse ao uso de transporte público como pontos de venda de bilhetes.
RF005	Como administrador da empresa preciso visualizar reclamações e sugestões advindas dos passageiros que enviaram por meio do aplicativo.

Fonte: Autor



Na Tabela 3 são descritas em formato de histórias de usuário as funcionalidades essenciais para o gerenciamento do serviço de transporte público por parte da prestadora de serviço, entretanto apenas os requisitos que são importantes para o desenvolvimento da solução proposta nesse trabalho estão sendo listados.

### Passageiro

A seguir é possível observar a Tabela 4 onde são colocados em evidência os principais requisitos para o funcionamento do aplicativo móvel para o passageiro, com o intuito de abordar os problemas que foram encontrados na pesquisa que pode ser melhor observada no Capítulo 4.

Tabela 4 – Requisitos Funcionais do Passageiro

ID	História de Usuário
RF001	Como passageiro preciso consultar as linhas disponíveis para a cidade de Picos.
RF002	Como passageiro preciso visualizar os horários e dias disponíveis para cada linha.
RF003	Como passageiro preciso visualizar em um mapa o percurso e os pontos de parada para uma determinada linha.
RF004	Como passageiro preciso visualizar as paradas mais próximas para entrar em um ônibus que esteja na linha desejada.
RF005	Como passageiro preciso visualizar a minha posição atual no mapa para melhor me localizar em relação as paradas.
RF006	Como passageiro preciso efetuar uma reclamação para as autoridades competentes por meio do aplicativo.
RF007	Como passageiro preciso registrar minhas linhas favoritas, para facilitar o meu acesso evitando visualizar informações que não são de meu interesse.
RF008	Como passageiro preciso visualizar as linhas disponíveis e os horários até quando não houver acesso a <i>internet</i> disponível.

Fonte: Autor

### 5.1.2 Requisitos Não Funcionais

Nesta seção serão descritos os principais requisitos não funcionais para obter uma melhor satisfação dos usuários em relação a utilização de transporte público, bem como restringir algumas tecnologias e comportamentos específicos para o desenvolvimento da aplicação, objetivando os recursos que cada tecnologia oferece em prol deste trabalho. Todavia os requisitos serão exibidos de forma categorizada com o intuito de melhorar o entendimento sobre os requisitos em cada ambiente, separando assim os requisitos em os que possuem relação com a aplicação dos administradores da empresa prestadora de serviço de transporte público e os referentes ao sistema de uso por parte dos seus passageiros.

#### Empresa de Transporte Público

Na Tabela 5 são descritas em formato de histórias de usuário as funcionalidades essenciais para o gerenciamento do serviço de transporte público por parte da prestadora de serviço, entretanto apenas os requisitos que são importantes para o desenvolvimento da solução proposta nesse trabalho estão sendo listados.

Tabela 5 – Requisitos Não Funcionais do Administrador da Empresa de Transporte Público

ID	Descrição
RNF001	O sistema deve solicitar autenticação de usuário e senha para obter acesso.
RNF002	O sistema deve funcionar para o ambiente <i>web</i> de modo as ser compatível com os principais navegadores, como: Chrome, Firefox, Safari e <i>internet Explorer</i> .
RNF003	O sistema deve estar disponível 24 horas por dia e 7 dias por semana.
RNF004	O sistema deve utilizar o sistema gerenciador de banco de dados PostgreSQL com a extensão PostGIS.
RNF005	O sistema deve utilizar a linguagem de programação Python em conjunto com o <i>framework</i> Django e sua biblioteca GeoDjango.

Fonte: Autor

#### Passageiro

Na Tabela 6 são descritas em formato de histórias de usuário as funcionalidades essenciais para o gerenciamento do serviço de transporte público por parte da

prestadora de serviço, entretanto apenas os requisitos que são importantes para o desenvolvimento da solução proposta nesse trabalho estão sendo listados.

Tabela 6 – Requisitos Não Funcionais do Passageiro

ID	Descrição
RNF001	O aplicativo do passageiro deve funcionar para a plataforma Android, compatível com as versões acima da 4.4 KitKat.
RNF002	O sistema deve possuir uma API para consultas de informações, utilizando a arquitetura REST e o formato de dados JSON.
RNF003	O aplicativo deve armazenar os dados consultados à API utilizando o banco de dados pra disponibilizar os dados quando não houver conexão com a <i>internet</i> .
RNF004	A API deve estar disponível 24 horas por dia e 7 dias por semana.
RNF005	A API deve suportar uma carga de acesso acima de 100 acessos simultâneos e mais de 10 mil consultas diárias.
RNF006	O aplicativo deve funcionar de forma nativa a plataforma Android utilizando a linguagem de programação Java.
RNF007	O aplicativo deve seguir as diretrizes de interface do usuário descritas no Material Design <sup>1</sup> .

Fonte: Autor

### 5.1.3 Modelagem do Sistema

Ao término do levantamento de requisitos de qualquer *software*, é impreterível a modelagem do sistema em virtude da dificuldade de entendimento das funcionalidades levantadas e, ao efetuar a modelagem do sistema facilita a compreensão do sistema por parte da equipe, bem como do próprio cliente, uma vez que comumente essa modelagem é feita por meio de diagramas UML que possibilitam uma visão gráfica do sistema, e não apenas textual.

Neste capítulo será possível visualizar dois dos principais diagramas da UML, o diagrama de classes e o diagrama de casos de uso. Tendo em vista a melhoria do entendimento reduziu-se os diagramas para demonstrar apenas o essencial para a compreensão da solução proposta por esse trabalho.

<sup>1</sup> <https://material.io/guidelines/>

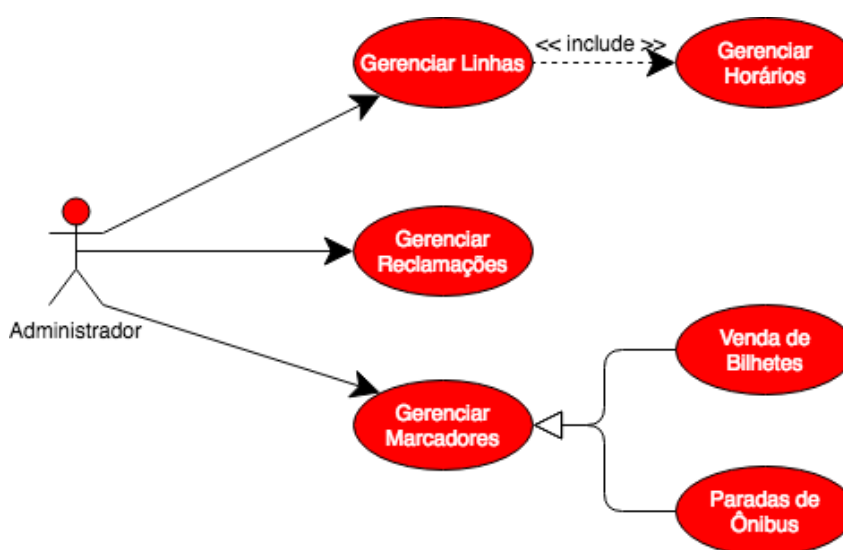
### 5.1.3.1 Diagrama de Casos de Uso

#### Empresa de Transporte Público

Na análise efetuada no diagrama de casos de uso da Figura 12 é possível visualizar que o administrador da empresa é o responsável por garantir o pleno funcionamento da aplicação, pois ele teria a responsabilidade de gerenciar os dados mais importantes do sistema, contudo ainda se deve colocar em destaque três desses casos de uso:

- **Gerenciar Linhas:** Consiste na manutenção das linhas atuais em que a empresa presta seus serviços. O administrador tem o dever de registrar as coordenadas geográficas referentes ao percurso do ônibus, bem como os seus horários de funcionamento em cada dia da semana.
- **Gerenciar Marcadores:** Possibilita cadastrar pontos importantes para a melhor prestação do serviço, pois é a funcionalidade que permite que os administradores cadastrem as paradas de ônibus e os locais de venda de bilhetes para o uso do transporte público, bem como a localização desses pontos no mapa.
- **Gerenciar Reclamações:** Permite que os administradores visualizem as reclamações efetuadas pelos passageiros por meio do aplicativo, assim, tendo a possibilidade de analisar de forma efetiva os problemas que ocorrem advindos do uso de transporte público.

Figura 12 – Diagrama de Casos de Uso do Administrador



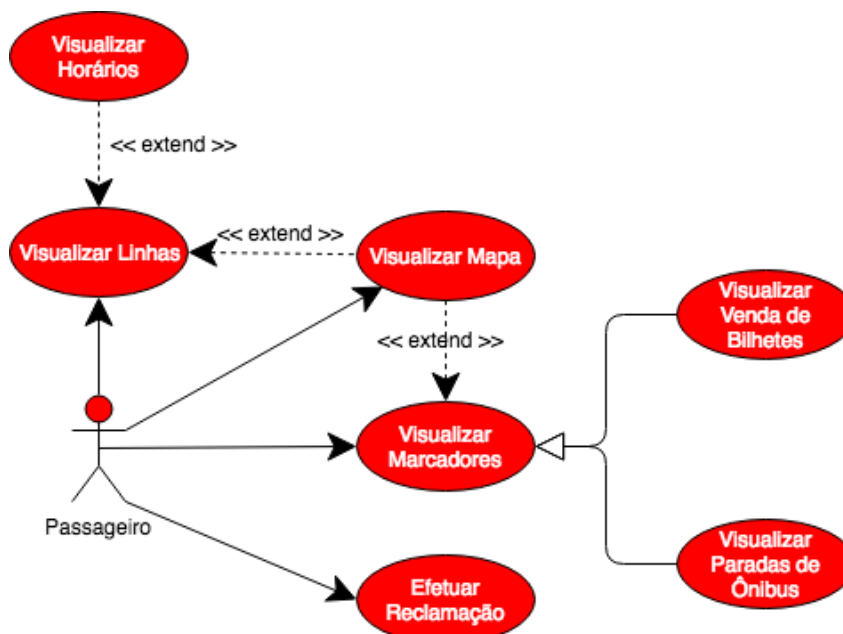
Fonte: Autor

## Passageiro

O problema em torno deste trabalho visa a melhoria da experiência do passageiro para com o uso do transporte público. Após todo o estudo focado no problema em questão foi possível desenvolver o diagrama de casos de uso presente na Figura 13, que demonstra as funcionalidades básicas disponíveis para os passageiros baseadas nos seguintes casos de uso:

- **Visualizar Linhas:** Possibilita ao passageiro visualizar uma lista de todas os percursos disponíveis para a cidade, bem como facilita o acesso a todos os horários disponíveis de cada percurso e, também, é permitido ao passageiro visualizar todo o percurso em um mapa.
- **Visualizar Marcadores:** Consiste na disponibilização de uma interface onde é possível observar uma lista com todos os pontos importantes para o uso de transporte público, como por exemplo: paradas de ônibus e pontos de vendas de bilhetes. Além disso, também é possível visualizar a posição desses marcadores em um mapa.
- **Efetuar Reclamação:** Permite aos passageiros de transporte público contatarem diretamente as pessoas responsáveis pelo transporte público, por meio de um formulário os usuários iram efetuar reclamações sobre a prestação do serviço de transporte público da cidade.

Figura 13 – Diagrama de Casos de Uso do Passageiro

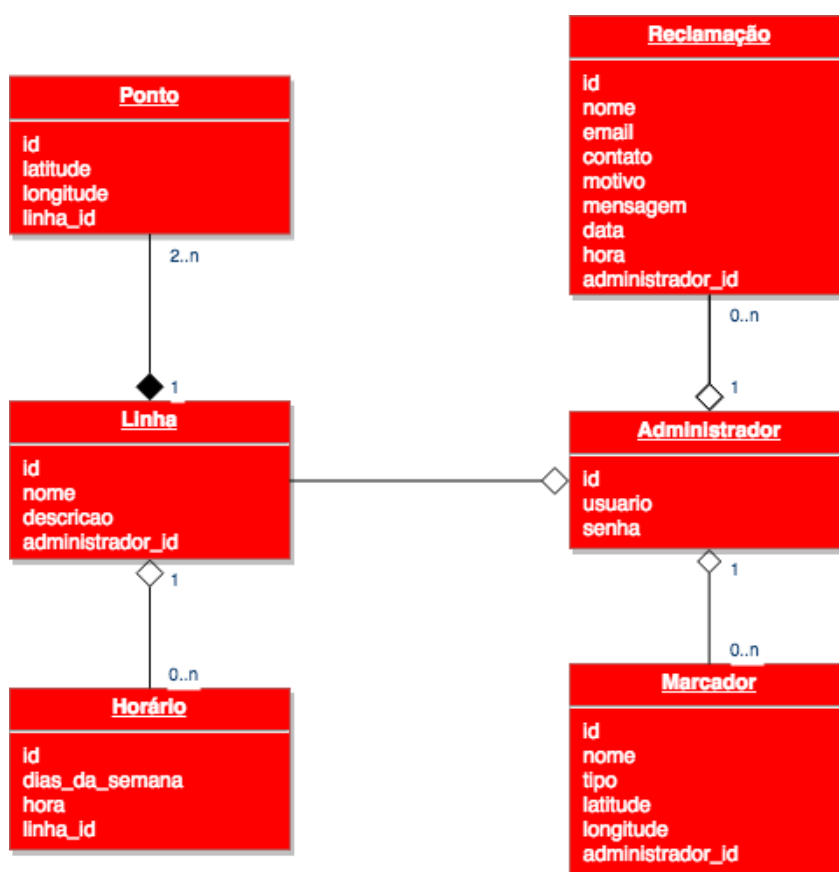


Fonte: Autor

### 5.1.3.2 Modelo Entidade-Relacionamento

Como pode ser visto na Figura 14 o modelo de banco de dados foi criado de forma genérica para atender mais de uma empresa. A empresa será representada por meio da entidade do administrador, em que possibilita que todas as demais entidades estão relacionadas direta ou indiretamente com a empresa, assim, tornando possível que cada empresa gerencie apenas os dados referentes a ela. Este modelo também reduz algumas entidades e atributos adicionais que são utilizadas para o melhor funcionamento da solução, contudo não se fazem necessários para o entendimento do que será desenvolvido.

Figura 14 – Diagrama Entidade-Relacionamento



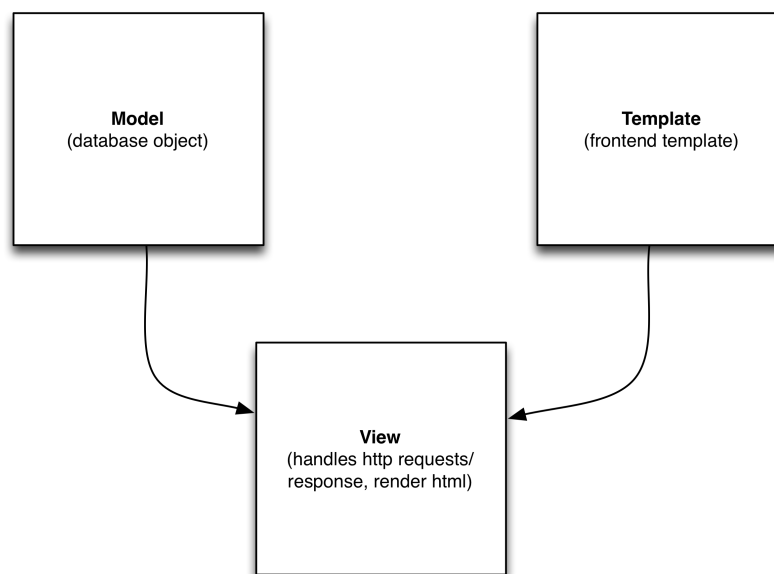
Fonte: Autor

## 5.2 ARQUITETURA DO SOFTWARE

### 5.2.1 Aplicativo Web

Para o desenvolvimento da versão *web* do aplicativo Topin, composta por um ambiente administrativo que possibilita o registro das informações necessárias para o funcionamento do sistema, foi utilizada a arquitetura de *software* MTV que como pode ser visto na Figura 15 é dividida em três camadas:

Figura 15 – Arquitetura Model-Template-View



Fonte: Cornell University

- **Model** é a única e definitiva fonte de acesso aos dados armazenados no banco de dados. Ele possui todos os campos e comportamentos essenciais dos dados que você está gerenciando no banco de dados.
- **Template** é o meio de apresentação dos dados requisitados pelo usuário, geralmente esses dados são apresentados em uma página HTML ou em uma formatação de dados como JSON ou XML.
- **View** é responsável pelas regras de negócio do sistema, bem como efetua o tratamento dos dados antes de serem enviados ao *template*, ou seja, quando um usuário solicita algum serviço por meio de uma URL, a *view* é quem deve efetuar os passos necessários para que o serviço seja executado.

### 5.2.2 Aplicativo Móvel

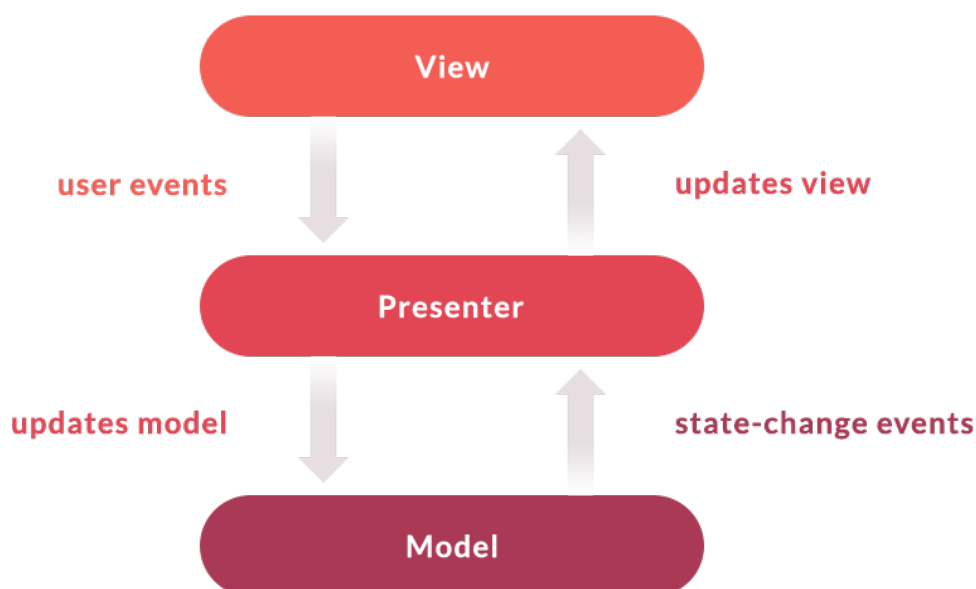
O aplicativo para dispositivos móveis que visa atender as necessidades dos usuários de transporte público foi desenvolvido para a plataforma Android utilizando o padrão de arquitetura de *software* MVP que como pode ser visto na Figura 16 é composto de três camadas:

- **Model** é a representação abstrata das informações que serão armazenadas, sendo esta a camada que demonstra os campos que serão atualizadas pelo presenter quando houver modificações na view.
- **View** é a camada visível ao usuário final do sistema, não contendo nenhuma regra de negócio, estando responsável apenas pela exibição e coleta de

dados pelo usuário, disparando quando necessário a mudança de estado de informações ou eventos para o presenter, para que então o presenter altere o estado das informações, quando necessário, no banco de dados por meio do model ou então execute alguma ação com base na chamada de um evento.

- **Presenter** é o intermédio entre o model e a view, estando encarregado de atualizar a view quando o model for alterado e sincronizar o model em relação a view. Como também possui a responsabilidade de efetuar o tratamento das informações antes de submeter a qualquer uma das camadas, sendo esta a camada onde é empregada as regras de negócio.

Figura 16 – Arquitetura Model-View-Presenter



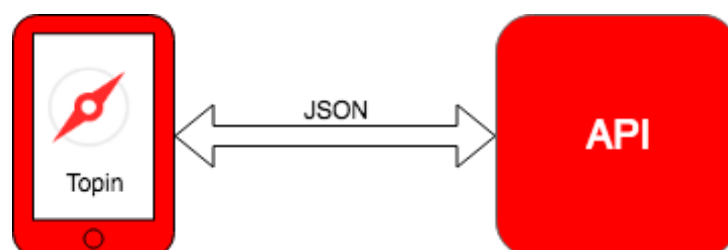
Fonte: Macroscope

### 5.2.3 API

O Topin possui duas aplicação independentes em seu modo de desenvolvimento, contudo existe a necessidade de uma interação constante entre as partes, para que possam interagir entre si é necessário o desenvolvimento de uma API, portanto foi determinado o uso da arquitetura REST, como pode ser visto na Figura 17 este modelo é composto de duas camadas, e para a comunicação entre as camadas foi determinado que os dados seriam transferidos no formato de dados JSON.



Figura 17 – Arquitetura REST



Fonte: Autor

Utilizando a arquitetura REST todos os dados disponíveis na API, serão acessados por meio de endereços eletrônicos com a utilização dos métodos corretos do protocolo HTTP, estes métodos podem ser vistos na Tabela 1 e, por meio da Tabela 7, é possível visualizar todos os recursos disponibilizados pela API do Topin.

Tabela 7 – Recursos Disponíveis na API

Método	URI	Utilização
GET	/api/markers	Recupera todos os marcadores.
GET	/api/markers/<id>	Recupera um marcador específico.
GET	/api/lines	Recupera todas as linhas disponíveis
GET	/api/lines/<id>	Recupera uma linha específica
GET	/api/lines/<id>/schedules	Recupera todos os horários de uma linha
POST	/api/complaint	Registra uma nova reclamação

Fonte: Autor

### 5.3 TECNOLOGIAS UTILIZADAS

Para a implementação do objeto deste trabalho foram utilizadas várias tecnologias, principalmente pela necessidade de desenvolvimento de três partes distintas que compõem o Topin, o sistema *web*, o aplicativo móvel e a API. A seguir serão demonstradas as tecnologias utilizadas para cada uma destas partes, entretanto como mencionado anteriormente o sistema *web* e a API funcionam de forma conjunta e são implementados em um único *software*, então as tecnologias utilizadas serão apresentadas de forma conjunta.

**Sistema Web e API**

Tabela 8 – Tecnologias Utilizadas no Sistema Web

<b>Tecnologia</b>	<b>Descrição</b>
Python <sup>2</sup>	Linguagem de programação orientada a objetos, interpretada e interativa.
Django <sup>3</sup>	O Django é um framework <i>web</i> gratuito, de código aberto e de alto nível que incentiva o rápido desenvolvimento e o design limpo e pragmático.
Django Rest Framework <sup>4</sup>	Módulo para o Django que possui um kit de ferramentas poderoso e flexível para criar API's com a arquitetura REST.
GeoDjango <sup>5</sup>	Módulo para o Django que o transforma em um framework <i>web</i> geográfico de classe mundial.
PostgreSQL <sup>6</sup>	Poderoso sistema de banco de dados objeto-relacional de código aberto.
PostGIS <sup>7</sup>	Extensão para o PostgreSQL que o transforma em um banco de dados geográfico.
GIT <sup>8</sup>	Sistema de controle de versão distribuído, gratuito e de código aberto projetado para lidar com projetos de pequeno e grande porte.

Fonte: Autor

<sup>2</sup> <https://www.python.org/><sup>3</sup> <https://www.djangoproject.com/><sup>4</sup> <https://www.django-rest-framework.org/><sup>5</sup> <https://docs.djangoproject.com/en/2.0/ref/contrib/gis/><sup>6</sup> <https://www.postgresql.org/><sup>7</sup> <https://postgis.net/><sup>8</sup> <https://git-scm.com/>

## Aplicativo Móvel

Tabela 9 – Tecnologias Utilizadas no Aplicativo Móvel

Tecnologia	Descrição
Java <sup>9</sup>	Linguagem de programação para desenvolvimento de aplicações em rede, móveis e incorporadas.
SQLite <sup>10</sup>	Mecanismo de banco de dados SQL autônomo, de alta confiabilidade, incorporado, repleto de recursos e de domínio público.
Room Persistence Library <sup>11</sup>	Biblioteca que fornece uma camada de abstração sobre o SQLite para permitir acesso fluente ao banco de dados, aproveitando todos os recursos do SQLite.
Retrofit2 <sup>12</sup>	Biblioteca para abstrair uma API REST em interfaces Java.
Google Maps API <sup>13</sup>	API que possibilita o acesso a todos os recursos do Google Maps.
GIT	Sistema de controle de versão distribuído, gratuito e de código aberto projetado para lidar com projetos de pequeno e grande porte.

Fonte: Autor

### 5.3.1 Python

Python é uma linguagem de programação de alto nível, no qual suporta vários paradigmas, como orientado a objetos, imperativo, funcional e procedural. É uma linguagem fortemente tipada, como também possui uma tipagem dinâmica. Um dos seus maiores diferenciais se deve ao fato de possuir um código de fácil leitura e exige a escrita de poucas linhas de código, caso seja comparado a programas, que possuam a mesma finalidade, desenvolvidos em outra linguagem. Como consequência de suas características, o uso desta linguagem se dá principalmente no uso de processamento de textos, dados científicos e desenvolvimento de páginas dinâmicas para a *web*.

<sup>9</sup> <https://www.java.com/>

<sup>10</sup> <https://www.sqlite.org/index.html>

<sup>11</sup> <https://developer.android.com/topic/libraries/architecture/room.html>

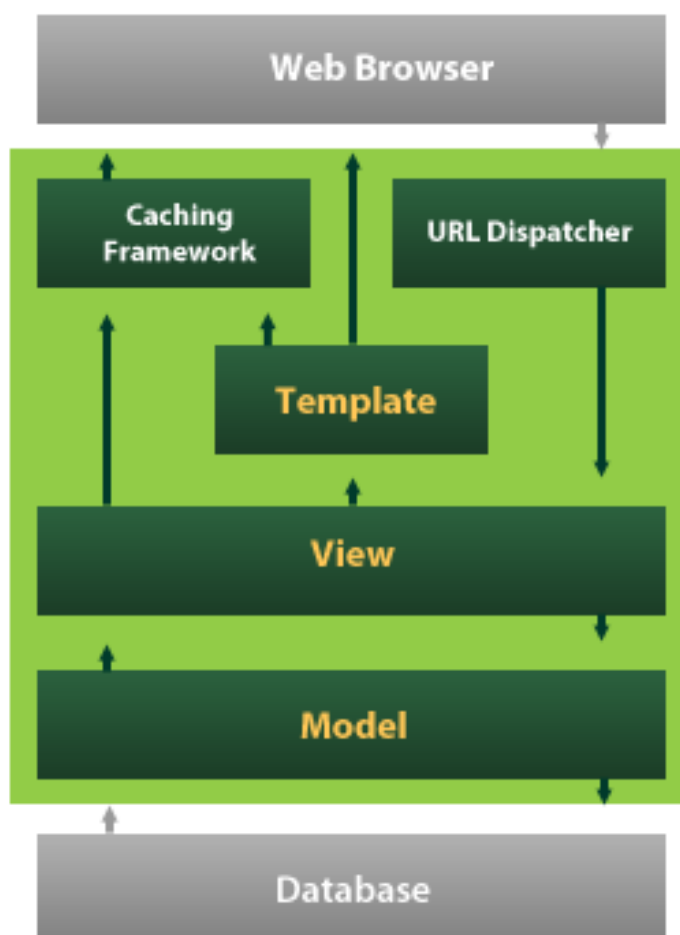
<sup>12</sup> <http://square.github.io/retrofit/>

<sup>13</sup> <https://developers.google.com/maps/?hl=pt-br>

### 5.3.2 Django

O Django é um framework de alto nível para a linguagem Python, que incentiva o desenvolvimento rápido, ao mesmo tempo que implementa um projeto de código simples e objetivo. Construído para que os desenvolvedores foquem no que precisa ser desenvolvido sem ter que reinventar a roda, o Django possui uma grande quantidade de mini frameworks embutidos para as mais diversas áreas do desenvolvimento *web*, como por exemplo a criação de API's, a construção de aplicações utilizando dados espaciais entre outros. A seguir são demonstradas algumas características principais que tornaram este framework a ferramenta necessário para o desenvolvimento do Topin.

Figura 18 – Arquitetura de Funcionamento do Django



Fonte: Chattyhive

#### 5.3.2.1 Django Rest Framework

O framework Django REST é um módulo que utilizado juntamente com o Django fornece um kit de ferramentas poderoso e flexível para criar APIs da *web*, pode-se

destacar algumas das principais razões para sua utilização no desenvolvimento de API's seguindo a arquitetura REST, são estas:

- A API navegável da *web*, fornecendo um meio gráfico e interativo para verificar o funcionamento da API.
- Políticas de autenticação, incluindo pacotes para o funcionamento de JWT, OAuth1 e OAuth2.
- Serialização que suporta fontes de dados ORM e não-ORM.
- Altamente personalizável, podendo apenas utilizar views baseadas em funções recursos rápidos e simples, a menos que necessário a utilização de recursos mais poderosos.
- Documentação extensa e excelente suporte da comunidade.

### 5.3.2.2 GeoDjango

GeoDjango é um módulo incluído por padrão no Django que o transforma em um framework *web* geográfico de classe mundial. O GeoDjango se esforça para tornar o mais simples possível a criação de aplicativos geográficos da *web*, como serviços baseados em localização. Suas características incluem:

- Tipos de campos para modelos do Django que abordam dados geométricos e geográficos.
- Extensões ao ORM do Django para consultar e manipular dados espaciais.
- Interfaces Python de alto nível e pouco acopladas para geometria GIS, bem como possibilita operações de rasterização e manipulação de dados em diferentes formatos.

### 5.3.3 PostgreSQL

O PostgreSQL é um sistema de gerenciamento de banco de dados objeto-relacional de código aberto. Possui suporte para todos os principais sistemas operacionais e atende fortemente as características ACID (Atomicidade, Consistência, Integridade e Durabilidade), como também inclui quase todos os tipos de dados: Integer, Numeric, Boolean, Char, Varchar, Date, Interval e Timestamp, bem como o armazenamento de objetos binários grandes e, com isso, suporta o armazenamento de imagens, sons ou vídeos, além de possuir uma boa documentação que fornece suporte a todos os recursos disponíveis.

### 5.3.3.1 PostGIS

O PostGIS é uma extensão que transforma o PostgreSQL em um banco de dados espacial. Ele adiciona suporte a objetos geográficos, permitindo que consultas de baseadas em localização sejam executadas no SQL. Além do reconhecimento básico de localização, o PostGIS também oferece:

- Funções analíticas e de processamento para dados vetoriais e raster;
- Álgebra de mapa com dados raster;
- Funções de reprojeção espacial para dados vetoriais e raster;
- Suporte para importação e exportação de dados vetoriais;
- Importação de dados raster;
- Renderização e importação de funções de suporte a dados vetoriais para formatos textuais;
- Renderização de dados raster;
- Funções de chamada SQL de dados raster e vetoriais sem costura para extrusão de valores de pixel por região geométrica, estatísticas de execução por região, recorte de rasters por uma geometria e rasters de vetorização;
- Suporte a objetos 3D, índice espacial e funções;
- Suporte a topologia de rede.

### 5.3.4 Java

O Java é uma linguagem de programação utilizada para o desenvolvimento de aplicações de todos os tipos, em rede, bem como é amplamente utilizada para a implementação de soluções móveis e embarcadas, como jogos, sistemas para a *internet* e softwares corporativos.

Esta linguagem foi projetada para permitir o desenvolvimento de aplicações multiplataforma de alto desempenho para os mais diversos ambientes de execução. Além de algumas dessas vantagens, o Java se tornou uma importante ferramenta de desenvolvimento, pois permite que os desenvolvedores:

- Armazenem um *software* em uma plataforma e o executem virtualmente em qualquer outro ambiente.
- Implementem softwares para a *internet*.
- Integre aplicações e serviços utilizando a linguagem para desenvolver aplicações personalizáveis.

- Desenvolva sistemas de alto desempenho e eficientes para celulares, processadores remotos, microcontroladores, módulos sem fio, sensores, *gateways*, produtos de consumo geral entre outros dispositivos.

### 5.3.5 SQLite

SQLite é uma biblioteca de código aberto e domínio público que implementa um banco de dados SQL transacional incorporado. Ao contrário da maioria dos outros bancos de dados SQL, o SQLite não possui um processo de servidor separado. O SQLite lê e grava diretamente em arquivos de disco comuns. Um banco de dados SQL completo com várias tabelas, índices, acionadores e visualizações está contido em um único arquivo de disco. O formato de arquivo de banco de dados é multi-plataforma. Esses recursos tornam o SQLite uma escolha popular como um formato de arquivo de aplicativo.

O SQLite é uma biblioteca compacta. Com todos os recursos ativados, o tamanho da biblioteca pode ser menor que 500 KB, dependendo da plataforma de destino e das configurações de otimização do compilador. Há uma compensação entre o uso de memória e a velocidade. O SQLite geralmente roda mais rápido quanto mais memória você der. No entanto, o desempenho geralmente é bom mesmo em ambientes com pouca memória. Dependendo de como é usado, o SQLite pode ser mais rápido que a E/S do sistema de arquivos direto.

O SQLite é testado com muito cuidado antes de cada lançamento e tem a reputação de ser muito confiável. A maior parte do código fonte do SQLite é dedicada apenas a testes e verificações. Um conjunto de testes automatizado executa milhões e milhões de casos de teste envolvendo centenas de milhões de instruções SQL individuais e atinge 100% de cobertura de teste de filial. O SQLite responde normalmente às falhas de alocação de memória e erros de E/S de disco. As transações são ACID mesmo que seja interrompido por falhas no sistema ou falhas de energia.

### 5.3.6 GIT

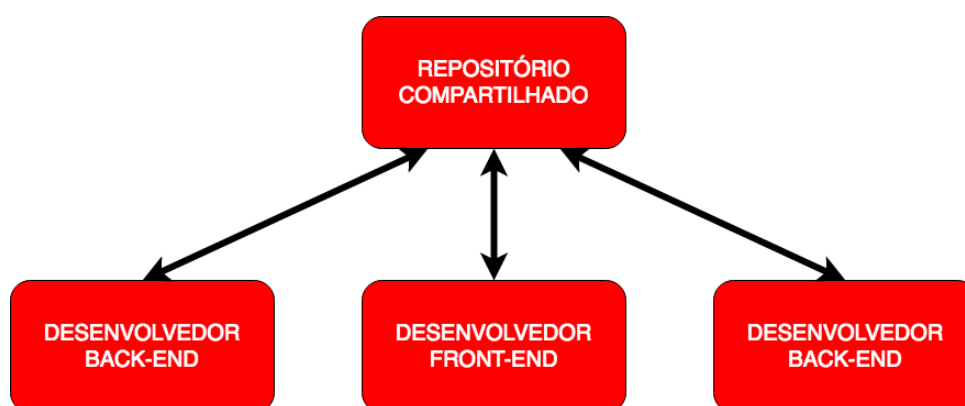
O Git é um sistema de controle de versão distribuído gratuito e de código aberto projetado para lidar desde projetos pequenos a projetos grandes e complexos, com velocidade e eficiência. As principais características do GIT são:

- **Ramificação e fusão:** Este recurso permite que o desenvolvedor tenha várias ramificações locais que podem ser totalmente independentes umas das outras.
- **Pequeno e Rápido:** Quase todas as operações são realizadas localmente, proporcionando uma enorme vantagem de velocidade em relação a sistemas

centralizados que constantemente precisam se comunicar com um servidor.

- **Distribuído:** Consiste em um sistema de controle de versão que, em vez de manter apenas a versão atualizada do código, ele copia na íntegra todo o repositório do servidor, incluindo as versões atuais e anteriores do sistema.
- **Integridade dos dados** O modelo de dados que o Git usa garante a integridade criptográfica de cada bit do seu projeto.
- **Área de preparação:** Área intermediária onde suas modificações podem ser formatadas e revisadas localmente antes de serem enviados ao repositório compartilhado.

Figura 19 – Arquitetura de Funcionamento do GIT



Fonte: Autor

## 5.4 FUNCIONALIDADES

Nesta sessão são apresentadas as principais funcionalidades do Topin. Para a efetivação desse trabalho foi desenvolvido dois módulos: sistema de gerenciamento da empresa de transporte público e o aplicativo móvel para os passageiros.

Para o entendimento das funcionalidades se faz necessária a compreensão de alguns aspectos do funcionamento do transporte público, como: linhas, trajetos, horários, marcadores e reclamações.

- **Linhas:** São as rotas disponibilizadas pela empresa de transporte coletivo, onde deve ser especificada a parada inicial e a parada final de cada linha.
- **Trajeto:** São as coordenadas geográficas de todos os pontos de referência, geralmente são coletadas principalmente as coordenadas geográficas de cada curva do caminho percorrido.
- **Horários:** São todos os horários de saída do veículo da origem das linhas disponíveis, com base na hora de partida e o dia da semana que este horário está disponível.



Figura 20 – Topin para Empresa - Tela de Cadastro da Linha

The screenshot shows the 'Topin' web application interface for company registration. The header includes the 'Topin' logo and user navigation links. The breadcrumb trail indicates the current path: 'Início > Route > Linhas > 01 Pantanal → Centro'. The main section is titled 'Modificar linha' and contains several form fields: 'Nome' (01 Pantanal → Centro), 'Identificador' (pantanal-centro), 'Descrição' (Pantanal -> Pedrinhas -> Centro), 'Cidade' (Picos), 'Administrador' (Dayvid Emerson), and 'Marcadores' (Parada do Pantanal). A 'HISTÓRICO' button is located in the top right corner. A footer note states: 'Mantenha pressionado o "Control", ou "Command" no Mac, para selecionar mais de uma opção.'

Fonte: Autor

- **Marcadores:** São todos os pontos de interesse do usuário, basicamente é composto por paradas, em que é possível pegar o transporte, e pontos de venda de bilhetes, onde é possível efetuar a compra de bilhetes para o uso do transporte público.
- **Reclamações:** É a mensagem deixada por um passageiro que tenha passado por algum inconveniente ou apenas deseja deixar seu comentário sobre sua experiência com o transporte público.

#### 5.4.1 Administradores de Empresa de Transporte Público

Os administradores da empresa de transporte público possuem um papel vital para o funcionamento do Topin, pois possuem a responsabilidade de registrar e gerir as informações que serão disponibilizadas aos passageiros, como pode ser visto na Figura 20 os administradores devem cadastrar todos os detalhes referentes as linhas dos transportes, bem como devem registrar os marcadores desta mesma linha, para que assim os passageiros visualizem apenas os marcadores que serão uteis no uso desta linha.

Ainda no cadastro das linhas é possível visualizar na Figura 21 que os administradores também deverão informar os horários disponíveis para o transporte público e, não apenas registrar os horários, como também separar estes horários por dia da semana, para que não exista a possibilidade do usuário ser informado de um horário

que não estará disponível no fim de semana, por exemplo.

Ao registrar os horários, a forma de registro dos dias da semana se destaca, pois para esta aplicação foi criado um campo no banco de dados PostgreSQL do tipo lista de valores, que se trata de um tipo de campo onde é permitido salvar um conjunto de valores sem que exista a necessidade criação de uma nova tabela para tal objetivo, por esse motivo o registro dos dias da semana para ser compatível com o banco de dados é colocado em uma cadeia de caracteres com os valores separados por vírgula, as opções de dias da semana são: *sunday, monday, tuesday, wednesday, thursday, friday, saturday*.

Ao visualizar a Figura 21 pode-se observar que além dos horários, o administrador irá registrar todos os pontos de uma linha, contudo estes pontos são referentes ao trajeto, ou seja, o caminho percorrido pelo transporte na linha em questão. Geralmente, os pontos de referência escolhidos são baseados em sua localização no mapa para formar o desenho do trajeto possível, como por exemplo são colocadas as coordenadas geográficas apenas das curvas do trajeto, para os perfeccionistas é possível registrar todas as coordenadas do trajeto, o que pode facilmente passar de 200 coordenadas geográficas.

Figura 21 – Topin para Empresa - Tela de Cadastro dos Horários e Trajeto da Linha

HORÁRIOS			
DIAS DA SEMANA	HORA	APAGAR?	
01 Pantanal → Centro - 06:00:00 sunday,monday,tuesday,v	06:00:00 Agora   🔍	<input type="checkbox"/>	
01 Pantanal → Centro - 06:20:00 monday,tuesday,wednes	06:20:00 Agora   🔍	<input type="checkbox"/>	
01 Pantanal → Centro - 06:40:00 monday,tuesday,wednes	06:40:00 Agora   🔍	<input type="checkbox"/>	
01 Pantanal → Centro - 07:00:00 sunday,monday,tuesday,v	07:00:00 Agora   🔍	<input type="checkbox"/>	
01 Pantanal → Centro - 07:30:00 monday,tuesday,wednes	07:30:00 Agora   🔍	<input type="checkbox"/>	
01 Pantanal → Centro - 08:00:00 sunday,monday,tuesday,v	08:00:00 Agora   🔍	<input type="checkbox"/>	
+ Adicionar outro(a) Horário			

PONTOS			
LONGITUDE	LATITUDE	ORDEM	APAGAR?
(-7.07177900, -41.39564100) -41,39564100	-7,07177900	1	<input type="checkbox"/>
(-7.07317600, -41.39712600) -41,39712600	-7,07317600	2	<input type="checkbox"/>
(-7.07279400, -41.39880700) -41,39880700	-7,07279400	3	<input type="checkbox"/>

Fonte: Autor

Os marcadores são pontos de interesse no uso de transporte público e para os administradores é disponibilizado um meio de registro desses pontos, ao analisar a Figura 22 é perceptível a possibilidade do administrador da empresa registrar a

localização do marcador, bem como é possível informar uma descrição do marcador no intuito de informar ao passageiros mais detalhes sobre o ponto registrado.

Figura 22 – Topin para Empresa - Tela de Cadastro dos Marcadores

Topin

BEM-VINDO(A), TOPIN. VER O SITE / ALTERAR SENHA / ENCERRAR SESSÃO

Início > Route > Marcadores > Parada do Pantanal

Modificar marcador

HISTÓRICO

Longitude: -41.39565700

Latitude: -7.07171700

Título: Parada do Pantanal

Identificador: parada-do-pantanal

Tipo: Parada de ônibus

Descrição:

Cidade: Picos

Administrador: Dayvid Emerson <dayvidemerson96@gmail.com>

Apagar

Salvar e adicionar outro(a)

Salvar e continuar editando

SALVAR

Fonte: Autor

Um dos principais problemas descritos nesse trabalho se deve a dificuldade dos passageiros entrarem em contato com as pessoas responsáveis pelo transporte público para descrever a sua indignação com o serviço prestado de alguma forma, como já foi demonstrado neste projeto os passageiros, muitas vezes, não sabem a quem reclamar e acabam culpando pessoas ou instituições que não tem autoridade para solucionar o problema encontrado.

Como pode ser visto na Figura 23, visando este problema foi disponibilizada aos administradores das empresas de transporte público uma seção onde eles podem acompanhar as reclamações que os passageiros enviaram por meio do aplicativo Topin.

Figura 23 – Topin para Empresa - Tela de Visualização das Reclamações



Fonte: Autor

#### 5.4.2 Passageiro

O aplicativo criado para o uso dos passageiros de transporte coletivo possui o objetivo de informá-los sobre detalhes referentes ao funcionamento do serviço de transporte público, porém antes de explicar as funcionalidades do aplicativo vale ressaltar que o aplicativo permite a visualização das informações sem a necessidade do uso de *internet*, por ser um recurso que funciona em segundo plano não existe a possibilidade de demonstrá-lo graficamente.

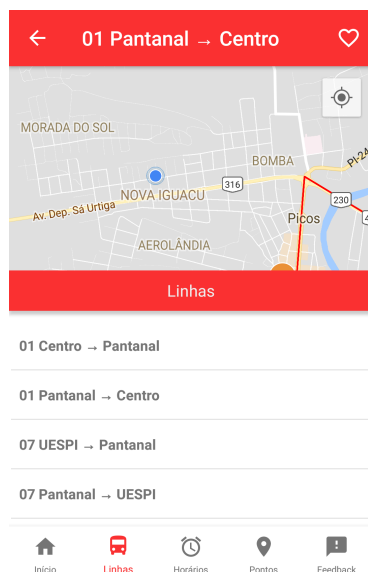
Entretanto todos os dados mostrados no aplicativo são consultados na API do Topin, em que necessita do uso de uma rede de dados, contudo o aplicativo ao recebê-los automaticamente os salva no banco de dados do *smartphone*, permitindo verificar em um acesso posterior se há *internet* disponível no dispositivo ou se foi encontrado algum erro na consulta das informações, caso isso aconteça, serão exibidos os dados armazenados no *smartphone* e, com isso, garantir que os passageiros terão acesso as informações que precisa mesmo que não possua *internet* disponível no momento.

Uma das funcionalidades principais do aplicativo pode ser vista na Figura 24 onde é possível perceber a presença de um espaço que permite visualizar todas as linhas disponíveis para uso do transporte público na região, além de dividir essas informações com um mapa demonstrando sua localização atual.

Também é permitido ao usuário arrastar a lista das linhas para cima possibilitando a visualização apenas das linhas disponíveis ocupando todo o espaço disponível na tela do *smartphone* ou como demonstrado na Figura 25 é possível deslizar a lista de linhas para baixo fazendo com que o mapa preencha toda a tela do aparelho, portanto o passageiro poderá ter uma visualização ampla da sua posição atual em relação as

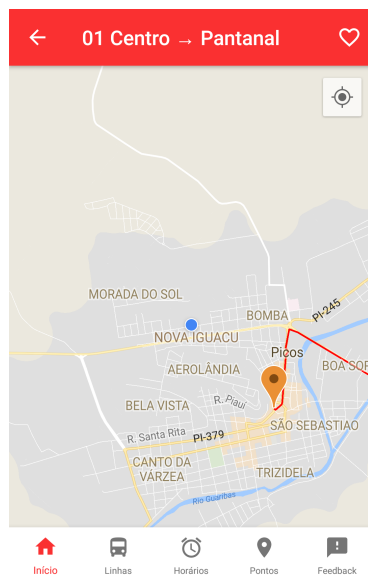
paradas e o trajeto da linha selecionada.

Figura 24 – Topin para Passageiros - Tela de Visualização das Linhas Disponíveis



Fonte: Autor

Figura 25 – Topin para Passageiros - Tela de Visualização da Linha Selecionada

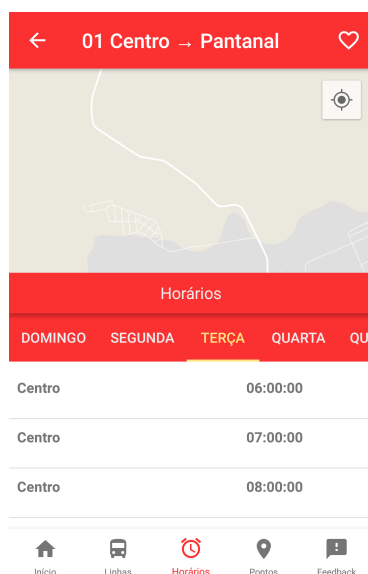


Fonte: Autor

Ao observar a Figura 26 é perceptível a presença do mesmo recurso existente para a lista de linhas, recurso este que possibilita a lista de horários ser arrastada tanto para cima, como para baixo, para destacar a visualização de um dos componentes da tela que está dividida.

O passageiro ao consultar a lista de horários automaticamente é demonstrado os horários do dia da semana baseado na data atual do *smartphone*, no entanto é permitido ao usuário alternar entre os dias da semana, que ao selecionar um dia da semana diferente será mostrado apenas o ponto de partida e os horários disponíveis para aquele dia da semana específico, como por exemplo a terça-feira que está selecionada na Figura 26 é possível visualizar todos os horários disponíveis e a origem a partir do ponto de partida da linha, nesse caso específico, a parada do Centro.

Figura 26 – Topin para Passageiros - Tela de Visualização dos Horários da Linha

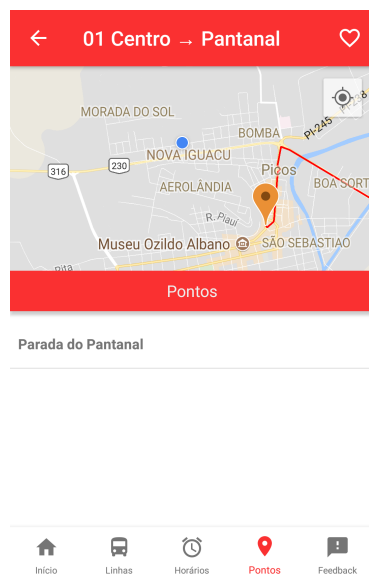


Fonte: Autor

Pode-se visualizar na Figura 27 que é possível visualizar os marcadores de interesse para a linha selecionada, em que é demonstrado no mapa por meio de um ícone todos os marcadores listados, ou seja, é visível não apenas a lista dos pontos de interesses, como também a sua geolocalização no mapa, e por meio do uso de sua localização atual, os passageiros podem se deslocar com uma maior facilidade para o ponto desejado.

Como dito anteriormente nesse trabalho uma das dificuldades encontradas pelos passageiros é a entrega de reclamações para as pessoas responsáveis pela prestação do serviço. Para alcançar esse objetivo é disponibilizado aos usuários do aplicativo móvel um ambiente (Figura 28) em que é possível enviar uma mensagem que será entregue as autoridades que competem a responsabilidade pela boa ou má prestação do serviço de transporte público.

Figura 27 – Topin para Passageiros - Tela de Visualização dos Pontos da Linha



Fonte: Autor

Figura 28 – Topin para Passageiros - Tela de Registro de Feedback

The screenshot displays the 'Feedback' screen in the Topin app. At the top, a red header bar contains a back arrow, the route '01 Centro → Pantanal', and a heart icon. Below the header is a red bar with the text 'Feedback'. The main area contains five input fields: 'Nome', 'E-mail', 'Contato', 'Motivo', and 'Mensagem'. Below these fields is a yellow button labeled 'ENVIAR'. At the bottom, a navigation bar includes icons for 'Início', 'Linhas', 'Horários', 'Pontos', and 'Feedback' (highlighted in red).

Fonte: Autor

## 6 CONSIDERAÇÕES FINAIS

Este capítulo apresenta os objetivos atingidos com o desenvolvimento do projeto em questão, bem como demonstra futuras melhorias que podem vir a ser desenvolvidas para a continuação deste trabalho.

### 6.1 CONCLUSÃO

No presente trabalho foi demonstrado todo o processo utilizado para o desenvolvimento da tecnologia utilizada para atender os principais problemas dos passageiros de transporte público da cidade de Picos, onde atualmente não existe nenhum meio de consulta a essas informações, não existe aplicativo móvel, site ou rede social com essas informações, o que ocasiona em problemas quanto a necessidade do usuário de se informar quanto ao uso do transporte público.

Com a finalidade de solucionar o problema em questão foi desenvolvido um sistema que com o auxílio das empresas de transporte público permite por meio de um *software*, nomeado de Topin, que as empresas gerenciem os dados referentes a prestação do serviço, bem como garante aos passageiros o acesso as informações sobre o uso do transporte público de forma rápida e fácil ao utilizar um aplicativo móvel.

O principal problema encontrado foi o método de coleta das coordenadas geográficas que possibilitam a construção do caminho percorrido pelos transportes para a disponibilização do trajeto no aplicativo, que muitas vezes os motoristas cortavam alguns caminhos que eles devem ir por que dependendo do horário eles presumiam que não haviam ninguém, então preferiam cortar caminho, contudo este problema contou com a ajuda da secretaria de transporte e as autoridades da empresa de transporte público, para evitar tais problemas e o aplicativo se mantivesse correto quanto as informações.

De acordo com as funcionalidades que vieram a ser desenvolvidas conclui-se que o sistema atende as necessidades básicas dos passageiros de transporte público, bem como o gerenciamento das informações das empresas de transporte público e facilita a entrega da mudança dessas informações aos passageiros, porém existem algumas melhorias que podem ser visualizadas no tópico a seguir.

### 6.2 TRABALHOS FUTUROS

Os trabalhos futuros se referem as atividades a serem executadas em um futuro próximo. Destas atividades as que mais se destacam são:



- Comparar o impacto na experiência dos passageiros antes e depois do uso do aplicativo desenvolvido.
- Desenvolver um módulo que permita o acompanhamento em tempo real dos transportes coletivos;

## REFERÊNCIAS

- Agência Brasil. *Smartphone se consolida como meio preferido de acesso à internet*. 2017. Disponível em: <<https://goo.gl/tm8Epn>>. Acesso em: 22 jul. 2017.
- ALVARADO, A. H. R.; JÚNIOR, N. F. R. C. Sistemas inteligentes de transporte aplicados em sistemas de alta capacidade: O caso brt e metrô da cidade de Brasília. In: *Congresso de Pesquisa e Ensino em Transporte*. Rio de Janeiro: Anpet, 2016. p. 1–11.
- BASS, L.; CLEMENTS, P.; KAZMAN, R. Software architecture in practice. *Boston, Massachusetts Addison*, 2003.
- BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. *UML: guia do usuário*. CAMPUS - RJ, 2006. ISBN 9788535217841. Disponível em: <<https://books.google.com.br/books?id=ddWqxcDKGF8C>>.
- CYSNEIROS, L. M.; LEITE, J. Definindo requisitos não funcionais. *XI Simpósio Brasileiro de Engenharia de Software*. Fortaleza, CE, p. 33, 1997.
- DANTAS, A.; TACO, P.; YAMASHITA, Y. Sistemas de informação geográfica em transportes: O estudo do estado da arte. In: ANPET BRASILIA. *Anais do X Congresso de Pesquisa e Ensino em Transportes*. [S.l.], 1996.
- FIELDING, R. et al. *Hypertext transfer protocol–HTTP/1.1*. [S.l.], 1999.
- FIELDING, R. T.; TAYLOR, R. N. Principled design of the modern web architecture. *ACM Transactions on Internet Technology (TOIT)*, ACM, v. 2, n. 2, p. 115–150, 2002.
- FIGUEIREDO, L. M. B. et al. Sistemas inteligentes de transporte. 2005.
- GUEDES, G. *UML 2 - Uma Abordagem Prática*. Novatec Editora, 2018. ISBN 9788575226445. Disponível em: <<https://books.google.com.br/books?id=mJxMDwAAQBAJ>>.
- HEUSER, C. *Projeto de banco de dados : Volume 4 da Série Livros didáticos informática UFRGS*. Bookman, 2009. ISBN 9788577804528. Disponível em: <[https://books.google.com.br/books?id=UKtB7\\_MnWQMC](https://books.google.com.br/books?id=UKtB7_MnWQMC)>.
- IBGE. *Censo demográfico*. 2010. Disponível em: <<http://www.censo2010.ibge.gov.br/sinopse/index.php?uf=22>>. Acesso em: 22 jul. 2017.
- IEEE. Standard glossary of softwareengineering terminology. *IEEE Software Engineering Standards & Collection*. IEEE, p. 610–12, 1990.
- ISO/TR 10992. *Intelligent transport systems — Use of nomadic and portable devices to support ITS service and multimedia provision in vehicles*. [S.l.], 2012.
- KNIBERG, H.; SKARIN, M. *Kanban and Scrum - Making the Most of Both*. C4Media Incorporated, 2010. (Enterprise software development series). ISBN 9780557138326. Disponível em: <<https://books.google.com.br/books?id=Hx1KAgAAQBAJ>>.

- MASSE, M. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. O'Reilly Media, 2011. ISBN 9781449319908. Disponível em: <<https://books.google.com.br/books?id=eABpzyTcJNIC>>.
- NASCIMENTO, C. A. D.; LIMA, J. P. O papel dos sistemas de informação como ferramenta para buscar melhorias na gestão dos serviços prestados ao usuário do transporte coletivo. *Revista Ciências Exatas*, Taubaté, v. 23, p. 1–8, 2017.
- NTU. *Sistemas inteligentes de transporte - its*. Brasília, 2013.
- PAULA FILHO, W. d. P. *Engenharia de software*. [S.l.]: LTC, 2003. v. 2.
- PRESSMAN, R. *Software Engineering: A Practitioner's Approach*. Boston, 2005. (McGraw-Hill higher education). ISBN 9780073019338. Disponível em: <<https://books.google.com.br/books?id=bL7QZHtWvaUC>>.
- REVISTA OPARA. *Ciências contemporâneas aplicadas*. Petrolina: [s.n.], 2014. v. 4.
- REZENDE, D. A. *Engenharia de software e sistemas de informação*. [S.l.]: Brasport, 2005.
- ROSE, A. *Uma avaliação comparativa de alguns sistemas de informação geográfica aplicados aos transportes*. Tese (Doutorado) — Universidade de São Paulo, 2001.
- SCHWABER, K. *Agile Project Management with Scrum*. Pearson Education, 2004. (Developer Best Practices). ISBN 9780735637900. Disponível em: <<https://books.google.com.br/books?id=6pZCAwAAQBAJ>>.
- SOMMERVILLE, I. *Software Engineering*. Pearson, 2011. (International Computer Science Series). ISBN 9780137053469. Disponível em: <<https://books.google.com.br/books?id=l0egcQAACAAJ>>.
- STAIR, R.; REYNOLDS, G. *Princípios de sistemas de informação*. Cengage Learning, 2011. ISBN 9788522107971. Disponível em: <[https://books.google.com.br/books?id=gNk\\\_SQAACAAJ](https://books.google.com.br/books?id=gNk\_SQAACAAJ)>.
- VÁZQUEZ, X. P. G. Sistemas de posicionamento global. *Eduga: revista galega do ensino*, Consellería de Educación e Ordenación Universitaria. Secretaría Xeral de Educación, n. 22, p. 153–158, 1999.