

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
CÂMPUS GUARAPUAVA
CURSO DE TECNOLOGIA EM SISTEMAS PARA INTERNET

GUSTAVO HENRIQUE PCHEK KWACZYNSKI

**WAYPARK: SISTEMA ELETRÔNICO DE PAGAMENTO PARA
ESTACIONAMENTO REGULAMENTADO**

TRABALHO DE CONCLUSÃO DE CURSO

GUARAPUAVA

2016

GUSTAVO HENRIQUE PCHEK KWACZYNSKI

**WAYPARK: SISTEMA ELETRÔNICO DE PAGAMENTO PARA
ESTACIONAMENTO REGULAMENTADO**

Trabalho de Conclusão de Curso apresentado ao
Curso de Tecnologia em Sistemas para Internet
da Universidade Tecnológica Federal do Paraná
como requisito parcial para a obtenção do título de
Tecnólogo.

Orientador: Prof. Dr. Roni Fabio Banaszewski

GUARAPUAVA

2016

**ATA DE DEFESA DE MONOGRAFIA DE TRABALHO DE CONCLUSÃO DE CURSO DO
CURSO DE TSI**

No dia 29 de novembro de 2016, às 16:30 horas, nas dependências da Universidade Tecnológica Federal do Paraná Câmpus Guarapuava, ocorreu a banca de **defesa da monografia** de Trabalho de Conclusão de Curso intitulada: **“WAYPARK: Sistema Eletrônico de Pagamento para Estacionamento Regulamentado”** do acadêmico **Gustavo Henrique Pchek** sob orientação do professor **Prof. Dr. Roni Fabio Banaszewski** do Curso de Tecnologia em Sistemas para Internet.

Banca Avaliadora	
Membro	Nome
Orientador	Prof. Dr. Roni Fabio Banaszewski
Coorientador	
Avaliador 1	Prof. Me. Carlos Eduardo Andrade Iatskiu
Avaliador 2	Prof. Esp. Fábio Leandro Janiszewski

Situação do Trabalho

Situação	<input checked="" type="checkbox"/> Aprovado <input type="checkbox"/> Aprovado com ressalvas <input type="checkbox"/> Reprovado <input type="checkbox"/> Não Compareceu
Encaminhamento do trabalho para biblioteca	<input checked="" type="checkbox"/> Pode ser encaminhado para biblioteca. <input type="checkbox"/> Manter sigilo para publicação ou geração de patente.

Guarapuava, 29 de novembro de 2016.

AGRADECIMENTOS

Agradeço a Deus pela força concedida, e também à minha família pelo suporte durante todo o período do Trabalho de Conclusão de Curso e da faculdade.

Agradeço ao meu orientador Prof. Dr. Roni Fabio Banaszewski pela brilhante orientação recebida e pelos conhecimentos transmitidos, os quais foram essenciais para a conclusão deste trabalho, e também aos professores do curso de Tecnologia em Sistemas para Internet pelo aprendizado durante todo o curso.

Agradeço às pessoas que se voluntariaram a responder as pesquisas e a realizar os testes ocorridos no decorrer deste trabalho, e também aos meus amigos que contribuíram de alguma forma para o desenvolvimento deste trabalho e que incentivaram-me a concluir este curso.

Gustavo Henrique Pchek Kwaczynski

RESUMO

PCHEK, Gustavo. WayPark: Sistema Eletrônico de Pagamento para Estacionamento Regulamentado. 53 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

Atualmente, a cobrança da permanência de veículos em estacionamentos regulamentados em várias cidades é comumente controlada por meio de cartões de papel. Em muitos casos, esta forma de controle se mostra ineficiente e pouco prática, devido ao tempo gasto comprando e preenchendo cartões, além da possibilidade de perda do cartão por erro de preenchimento e extravio de notificações ou multas recebidas. Deste modo, este projeto consiste no desenvolvimento de um sistema eletrônico para cobrança de estacionamentos regulamentados. O sistema proposto oferecerá uma área administrativa para gerenciamento do estacionamento; uma aplicação móvel para uso dos fiscais de trânsito em suas tarefas cotidianas de fiscalização e uma aplicação móvel para uso dos cidadãos que precisem pagar e gerenciar seus créditos para usufruir das vagas de estacionamento regulamentado. Com este sistema, busca-se promover uma maior agilidade e comodidade aos cidadãos na hora de estacionar seus veículos em vias públicas e também ao próprio órgão fiscalizador dos estacionamentos em cada cidade.

Palavras-chave: Estacionamento Regulamentado, Aplicativo, Sistema web

ABSTRACT

PCHEK, Gustavo. WayPark: Eletronic Payment System for Regulated Park. 53 f. Trabalho de Conclusão de Curso – Curso de Tecnologia em Sistemas para Internet, Universidade Tecnológica Federal do Paraná. Guarapuava, 2016.

Currently, the collection of regulated parking fees in many cities is generally controlled using paper cards. In many cases, this way of collecting proved to be inefficient and impractical, because of the time spent purchasing and filling out tickets, in addition to the possible loss of the ticket by fill error and loss of notifications or fines received. Thus, this project consists in the development of a electronic system for charging regulated parking. The proposed system offers an administrative area for parking management; a mobile application for parking inspectors for use in everyday tasks and a mobile application for citizens who have to pay and manage your credits to make use of regulated parking lots. With this system, it seeks to promote greater agility and convenience to citizens in time to park their vehicles on public roads and also to supervision body of car parks in every city.

Keywords: Regulated Park, App, Web System

LISTA DE FIGURAS

Figura 1	– Pango - Tela principal	13
Figura 2	– Mobilicidade	14
Figura 3	– Caixa do Produto - Visões frontal e traseira	23
Figura 4	– Modelagem de dados	24
Figura 5	– Wireframe da Página Principal	25
Figura 6	– Página Principal	29
Figura 7	– Arquitetura Comportamental da API	30
Figura 8	– Arquitetura Comportamental da Aplicação dos Clientes	32
Figura 9	– Página de Login	33
Figura 10	– Página de Cadastro	34
Figura 11	– Página Principal	35
Figura 12	– Menu e Página Minha Conta	36
Figura 13	– Modal Placa	37
Figura 14	– Páginas de Compra e Histórico de Compras	38
Figura 15	– Históricos de Tickets Ativados e Multas	39
Figura 16	– Página de Login	40
Figura 17	– Página Principal	41
Figura 18	– Menu Principal	42
Figura 19	– Respostas da pergunta: Você encontrou dificuldades para utilizar o aplicativo?	45
Figura 20	– Respostas da pergunta: Como você avalia a Interface Gráfica, em uma nota de 1 a 5?	45
Figura 21	– Respostas da questão: Selecione as funcionalidades que você mais gostou no aplicativo.	46
Figura 22	– Respostas da questão: Deste o cadastro até a ativação do ticket, você acha que:	46
Figura 23	– Respostas da pergunta: Você utilizaria o WayPark para pagar pelo uso de estacionamento, como alternativa aos cartões de papel?	46
Figura 24	– Respostas da pergunta: Você utiliza automóvel para se locomover pela cidade?	51
Figura 25	– Respostas da pergunta: Você encontra dificuldades em encontrar lugar para estacionar no centro da cidade?	52
Figura 26	– Respostas da pergunta: Numa escala de 1 a 5, como você avalia a forma de cobrança atual do estacionamento regulamentado em Guarapuava?	52
Figura 27	– Respostas da pergunta: Você já recebeu pontos na carteira por não saber que foi multado pelo EstaR?	52
Figura 28	– Respostas da pergunta: Você utilizaria um aplicativo para Smartphone para pagar pelo uso do estacionamento regulamentado?	53
Figura 29	– Respostas da pergunta: Quais das seguintes formas de cobrança você prefere?	53
Figura 30	– Respostas da pergunta: Quais destas formas de compra de créditos é a melhor para você?	53

LISTA DE SIGLAS

HTML	Linguagem para Marcação de Hipertexto (do inglês <i>HyperText Markup Language</i>)
CSS	Folhas de Estilo em Cascata (do inglês <i>Cascading Style Sheets</i>)
Sass	Folhas de Estilo Sintaticamente Impressionantes (do inglês <i>Syntatically Awesome Style Sheets</i>)
SQL	Linguagem de Consulta Estruturada (do inglês <i>Structured Query Language</i>)
API	Interface de Programação de Aplicativos (do inglês <i>Application Programming Interface</i>)
REST	Transferência de Estado Representacional (do inglês <i>Representational State Transfer</i>)
MVC	Modelo-Visão-Controlador (do inglês <i>Model-View-Controller</i>)
Haml	Linguagem de Marcação de Abstração HTML (do inglês <i>HTML Abstraction Markup Language</i>)
CORS	Compartilhamento de Recursos entre Origens (do inglês <i>Cross-Origin Resource Sharing</i>)
CRUD	Criar, Ler, Atualizar e Excluir (do inglês <i>Create, Read, Update and Delete</i>)
JSON	Notação de Objeto JavaScript (do inglês <i>Javascript Object Notation</i>)

SUMÁRIO

1	INTRODUÇÃO	9
1.1	OBJETIVOS	10
1.1.1	Objetivo Geral	10
1.1.2	Objetivos Específicos	10
1.1.3	Diferencial Tecnológico	11
2	RESENHA LITERÁRIA	12
2.1	ESTADO DA ARTE	12
2.1.1	Pango	12
2.1.2	Stop Fácil Estacionamento	13
2.1.3	Mobilicidade Estacionamento Regulamentado	13
2.1.4	Comparativo	14
2.2	FUNDAMENTAÇÃO TEÓRICA	15
2.2.1	HTML 5	15
2.2.2	CSS 3 e Sass	16
2.2.3	Javascript e Typescript	16
2.2.4	Angular	16
2.2.5	Material Design e Materialize	16
2.2.6	Cordova e Ionic 2	17
2.2.7	Ruby e Ruby On Rails	17
2.2.8	PostgreSQL	18
2.2.9	Scrum	18
3	METODOLOGIA	19
4	ANÁLISE E PROJETO	21
4.1	ANÁLISE DO SISTEMA	21
4.1.1	Histórias	21
4.1.2	Pesquisa	22
4.2	PROJETO DO SISTEMA	22
4.2.1	Caixa do Produto	23
4.2.2	Modelagem de Dados	24
4.2.3	Protótipos das Telas	24
4.3	CENÁRIOS	25
4.4	CONCLUSÃO	26
5	DESENVOLVIMENTO	27
5.1	ÁREA ADMINISTRATIVA E API	27
5.1.1	Área Administrativa	27
5.1.2	API	30
5.2	APLICAÇÃO DOS CLIENTES	31
5.3	APLICAÇÃO DOS AGENTES DE TRÂNSITO	40
5.4	CONCLUSÃO	43
6	RESULTADOS	44
7	CONSIDERAÇÕES FINAIS	47
7.1	FUTURAS ATUALIZAÇÕES	47

REFERÊNCIAS	49
Apêndice A – PESQUISA	51

1 INTRODUÇÃO

O estacionamento regulamentado é controlado em muitas cidades por meio de cartões de papel. Estes cartões são vendidos de forma avulsa ou em talões em pontos de vendas credenciados pelo órgão responsável em cada município. Basicamente, cada cartão equivale a um certo tempo de permissão para estacionar em vias públicas, geralmente na zona central da cidade. Atualmente, o manuseio dos cartões em muitas cidades brasileiras não é muito prático, pois além de exigir certo esforço para encontrar os pontos de vendas próximos ao local de estacionamento, ainda é preciso possuir uma caneta para marcar em um ou mais cartões a data e hora de chegada, dependendo do tempo desejado para permanência no local. Além disso, se faz necessário posicionar tais cartões no interior do veículo de forma que as anotações sejam facilmente identificadas pelos fiscais de trânsito. Como regra, os cartões são posicionados em cima do painel frontal do veículo para serem visíveis pelo para-brisa.

Os fiscais do estacionamento circulam pelas vias públicas, em zonas regulamentadas, a fim de verificar a existência, o correto preenchimento e o prazo de expiração dos cartões em cada veículo. Caso um fiscal encontre alguma irregularidade, este pode emitir uma multa ao condutor do veículo. Estas multas são preenchidas em um cartão de papel e normalmente são presas nos limpadores de para-brisa dos veículos, principalmente por conta da ausência dos condutores para notificação direta.

Esta forma de cobrança do estacionamento regulamentado é ineficiente sob alguns pontos de vista, mesmo quando o condutor já possui um cartão de estacionamento e esteja munido de caneta a fim de evitar a busca por um local credenciado de venda. Neste caso, ele precisa se atentar para preencher um ou mais cartões de forma correta para que sejam válidos para o período de tempo desejado. Caso haja desatenção, o ato de preenchimento manual indevido pode levar inclusive à inutilização de um cartão. Ademais, esta forma de cobrança demanda o uso excessivo de papel que além do impacto ambiental, ainda pode levar a alguns problemas de ordem física, tal como o extravio do cartão de multa fixado nos para-brisas dos veículos. Neste caso, a notificação emitida pelo fiscal pode não tomar o efeito esperado, ou seja, levar o motorista à ciência imediata de sua indisciplina. Este problema pode ser maior ainda se considerado

que se a multa do estacionamento regulamentado não for paga, o motorista recebe pontos na carteira de acordo com o Código Brasileiro de Trânsito.

Conforme as deficiências supracitadas, seria importante a existência de uma ferramenta computacional que permita automatizar estas práticas de controle e pudesse eliminar em médio prazo o uso de cartões de papel. De fato, uma solução computacional em forma de aplicativo móvel com acesso à Internet poderia resolver muitos dos problemas mencionados, oferecendo maior agilidade aos motoristas e controle aos órgãos fiscalizadores. Deste modo, a proposta que envolve este trabalho consistiu exatamente na construção de uma solução computacional para automatizar parte das atividades relacionadas a cobrança e gerenciamento de estacionamentos regulamentados.

A solução computacional desenvolvida neste trabalho é um protótipo que pode ser adaptado a diferentes estacionamentos regulamentados, resultando na otimização dos processos e melhora na confiabilidade e segurança dos registros de dados. Além disso, ela também fornece uma nova alternativa de pagamento para os clientes do estacionamento, aprimorando e fornecendo uma nova ferramenta de fiscalização para os agentes de trânsito e permitindo um amplo controle da movimentação financeira nos estacionamentos pelos administradores.

1.1 OBJETIVOS

1.1.1 OBJETIVO GERAL

O principal objetivo do trabalho é o desenvolvimento de um sistema online para gerenciamento e cobrança de estacionamentos regulamentados, destinado tanto para fiscais quanto para usuários.

1.1.2 OBJETIVOS ESPECÍFICOS

- Desenvolver um sistema web para controle de pagamento em estacionamentos regulamentados;
- Desenvolver um site para uso dos usuários do estacionamento regulamentado que permita compra, controle e utilização de créditos de estacionamento;
- Desenvolver um aplicativo para uso dos usuários do estacionamento regulamentado que permita compra, controle e utilização de créditos de estacionamento;
- Desenvolver um aplicativo para uso dos fiscais de trânsito;

- Realizar uma simulação do uso do sistema e avaliar os resultados.

1.1.3 DIFERENCIAL TECNOLÓGICO

O projeto busca formas de resolver os problemas existentes na regulamentação de estacionamentos que não possuam grande número de usuários, como pequenas e médias cidades, onde geralmente o controle do estacionamento é feito sem nenhum sistema computacional. Além disso, o projeto também busca desenvolver uma ferramenta que possua facilidade de uso e possa ser executada em computadores ou celulares sem alto poder de processamento.

Um dos diferenciais do projeto é a solução para os diferentes setores que envolvem o funcionamento dos estacionamentos: pagamento pelos clientes, fiscalização pelos agentes e controle pelos administradores. A ferramenta desenvolvida é adaptável aos diferentes estacionamentos sem que grandes alterações sejam necessárias. Além disso, a interface gráfica foi construída para que seja fácil e prática de utilizar tanto pelos clientes quanto pelos agentes e administradores do estacionamento. Os aplicativos desenvolvidos poderão ser exportados para Navegador Web, Android e iOS.

O nome escolhido - WayPark - une as palavras do idioma inglês *way*, que significa caminho ou maneira, e *park*, que significa estacionamento, referencia a atividade de estacionar pelo caminho ou também uma nova maneira de estacionar. Além disso, a pronúncia rápida das duas sílabas auxilia a divulgação e memorização.

2 RESENHA LITERÁRIA

A seguir serão apresentadas soluções que já foram implementadas para melhorar o controle de estacionamentos regulamentados tradicionais.

2.1 ESTADO DA ARTE

Nos últimos anos algumas alternativas automatizadas que substituem os tradicionais cartões de papel têm sido cada vez mais utilizadas. Dentre estas alternativas, destacam-se os sistemas que permitem o controle do usuário do estacionamento através de um aplicativo para *smartphone* e sistemas que gerenciam o processo de compra de cartões removendo o uso de papéis e planilhas pelos fiscais e agentes dos órgãos reguladores de estacionamento.

2.1.1 PANGO

Pango (abreviação de Pay, Park and Go – em português, pague, estacione e vá) é um aplicativo para *smartphone* com a finalidade de controlar estacionamentos regulamentados públicos ou privados. Através de uma conta criada no sistema do Pango, o usuário consegue controlar a quantia gasta nos estacionamentos (PANGO, 2014).

O aplicativo Pango foi desenvolvido por uma empresa israelense e está presente em vários países. Além da grande abrangência geográfica, o Pango pode ser utilizado em diferentes estacionamentos, como estacionamentos privados ou condomínios.

Atualmente no Brasil, apenas a cidade de Curitiba utiliza o controle de estacionamentos com o Pango (PANGO, 2014). Mesmo após a implantação em 2013, os cartões de papel continuaram a ser utilizados. A Figura 1 exibe a tela principal do aplicativo Pango.



Figura 1: Pango - Tela principal

2.1.2 STOP FÁCIL ESTACIONAMENTO

O Stop Fácil Estacionamento é um sistema que oferece gerenciamento completo de estacionamentos rotativos, com compra de créditos virtuais através de cartões de crédito (STOP-FACIL, 2014). Nesse sistema os usuários do estacionamento não possuem controle sobre os cartões / tíquetes ou créditos utilizados. Porém, os fiscais do estacionamento possuem um equipamento eletrônico, no qual podem debitar créditos de um determinado veículo, além de verificar sua regularidade (se o veículo possui créditos).

Em setembro de 2014 o sistema Stop Fácil foi implantado na cidade de União da Vitória, no Paraná, substituindo a forma de cobrança anterior (VVALE, 2014). Porém em julho de 2015 o sistema deixou de ser utilizado por conta de um desacordo entre a empresa e o órgão municipal (VVALE, 2015).

2.1.3 MOBILIDADE ESTACIONAMENTO REGULAMENTADO

O aplicativo chamado Estacionamento Eletrônico, desenvolvido pela empresa Mobilidade, assim como o Pango dispensa o uso de tíquetes em papel e permite a compra e ativação

de tíquetes de estacionamento através do aplicativo, além de outras funcionalidades como a geolocalização do veículo (MOBILICIDADE, 2015).

Atualmente, o Estacionamento Eletrônico é utilizado em 10 cidades brasileiras (MOBILICIDADE, 2015). Há uma versão diferente do aplicativo para cada cidade, que pode conter funcionalidades ou configurações específicas para que se adapte a determinada região.

A Figura 2 exibe as telas de login e principal de uma das versões do Mobilicidade Estacionamento Regulamentado.



Figura 2: Mobilicidade

2.1.4 COMPARATIVO

A Tabela 1 exibe uma comparação de recursos e funcionalidades disponíveis nas diferentes soluções supracitadas. Os recursos comparados são:

- **Acesso pelo usuário:** O usuário possui acesso direto ao sistema, através de um site ou aplicativo móvel;
- **Geograficamente abrangente:** Uma única instalação do sistema será necessária para controlar diferentes estacionamentos, sem necessidade de instalações separadas para cada estacionamento;
- **Consumo de créditos sem interação com o usuário:** A cobrança ou consumo de créditos é realizada sem que o usuário tenha controle direto sobre isso. Um exemplo é o consumo de créditos ativado pelos agentes de estacionamento;

- **Pagamento digital:** É possível comprar créditos online, sem a necessidade de adquiri-los em pontos de venda.
- **Versão web:** O usuário pode acessar suas informações no sistema através de um site.

Tabela 1: Comparativo

	Pango Brasil	Stopfacil Estacionamento	Mobilicidade Estacionamento Eletrônico	Sistema resultante deste projeto
Acesso pelo usuário	x		x	x
Geograficamente abrangente	x		Necessita de diferentes versões do aplicativo	
Consumo de créditos sem interação com o usuário		x		
Compra de créditos em pontos de venda		x		
Pagamento digital	x		x	x
Versão web	x		x	x

2.2 FUNDAMENTAÇÃO TEÓRICA

A seguir serão apresentadas as tecnologias e ferramentas que serão utilizadas no desenvolvimento do projeto. Foram definidas algumas linguagens de programação, *frameworks* e uma metodologia de desenvolvimento ágil.

2.2.1 HTML 5

HTML é uma sigla que significa Linguagem para Marcação de Hipertexto. O conceito de hipertexto pode ser entendido como todo o conteúdo inserido em um documento para a *web* que pode se interligar a outros documentos da *web* (SILVA, 2011).

Em sua quinta versão (HTML 5), o HTML é o tipo de conteúdo que mais trafega na internet, sendo utilizado em praticamente todos os websites existentes. Seus padrões são definidos pela W3C (World Wide Web Consortium), um consórcio formado por instituições comerciais e educacionais com o objetivo de definir padrões web (WILLIAM, 2012).

2.2.2 CSS 3 E SASS

CSS (Folhas de Estilo em Cascata) é uma ferramenta para adicionar estilos como fontes e cores aos documentos *web* e tem por finalidade devolver à marcação HTML/XML o propósito inicial da linguagem (SILVA, 2012). Sass é um pré-processador de CSS que permite o uso de funcionalidades não existentes no CSS como variáveis, aninhamento e herança. (SASS, 2016).

2.2.3 JAVASCRIPT E TYPESCRIPT

JavaScript é uma linguagem de programação que foi desenvolvida com a finalidade de fornecer um meio de adicionar maior interatividade e dinâmica a uma página *web* (SILVA, 2010).

TypeScript é um superconjunto tipado de que é compilado em JavaScript simples e apresenta muitas funcionalidades do ECMAScript 6. Permite aos desenvolvedores usarem ferramentas e práticas de alta produtividade como análise estática e refatoração de código enquanto desenvolvem aplicações JavaScript. (TYPESCRIPT, 2016).

2.2.4 ANGULAR

Angular (também chamado de Angular 2) é um *framework* JavaScript que simplifica o desenvolvimento de aplicações web dinâmicas e viabiliza a implementação do modelo MVC (Model-View-Controller). O Angular apresenta algumas características como desempenho, produtividade, fácil customização, além de ter fácil integração com outros *frameworks* e ferramentas JavaScript (PEREIRA, 2015).

Lançado em 2012 e desenvolvido por uma comunidade, com participação do Google, o Angular mostra-se em crescente ascensão devido aos conceitos que apresenta e à sua simplicidade de uso.

2.2.5 MATERIAL DESIGN E MATERIALIZE

Material Design é uma linguagem visual desenvolvida pela Google que sintetiza os princípios clássicos do bom design com a inovação e possibilidade de tecnologia e ciência. Em constante evolução e desenvolvimento, o projeto do Material Design tem como objetivo desenvolver um único sistema subjacente que permite uma experiência unificada em diferentes plataformas e objetivos. (GOOGLE, 2015).

Materialize é um *framework front-end* moderno e responsivo baseado no Material De-

sign. (MATERIALIZE, 2016). Fornece diversas ferramentas para a construção de templates como suporte a *grids* para layouts responsivos, campos de formulários, botões, paleta de cores, listas, *parallax*, entre outros.

2.2.6 CORDOVA E IONIC 2

Apache Cordova é um *framework* de desenvolvimento para dispositivos móveis de código aberto. Permite usar tecnologias *web* como HTML5, CSS3 e JavaScript para desenvolvimento multi-plataforma, evitando linguagens de desenvolvimento nativas de cada plataforma *mobile* (CORDOVA, 2015).

Ionic é um *framework* de desenvolvimento de aplicações móveis com foco no desenvolvimento híbrido. O Ionic 2 é a segunda versão do *framework*, construído em cima do Angular (Angular 2) e é uma reescrita completa da versão original (IONIC, 2016).

O Ionic 2 fornece várias ferramentas que auxiliam o desenvolvimento dos aplicativos. Elementos visuais como listas, botões, ícones e menus são disponibilizados para uso já formatados no visual do Material Design, e uma API com diversas funcionalidades como controladores e serviços para a manipulação dos componentes e funcionamento do aplicativo.

Devido à sua versatilidade por permitir a criação de aplicativos para diferentes plataformas e possibilidade de utilização de tecnologias conhecidas como o HTML5, JavaScript e CSS, o Ionic 2 foi utilizada como ferramenta para o desenvolvimento dos aplicativos deste projeto.

2.2.7 RUBY E RUBY ON RAILS

Ruby é uma linguagem de programação de código aberto com foco na simplicidade e na produtividade, que possui sintaxe de leitura natural e fácil escrita. Sendo uma das linguagens mais utilizadas no mundo e com crescente popularidade, Ruby é uma linguagem que equilibra a programação funcional com a programação interativa (RUBY, 2015).

A principal ferramenta que ajudou a popularidade do Ruby a crescer é o Ruby On Rails, que é um *framework web* que permite o desenvolvimento rápido de sites com código limpo e fácil manutenção. A utilização do Rails envolve uma forma diferente de pensar o desenvolvimento *web*, que prioriza a produtividade (JR; BARAZI, 2011).

Com o *framework* Ruby On Rails é possível criar diferentes aplicações web para os mais diversos fins, como websites, blogs, sistemas web, entre outros. Lançado há mais de uma década, o Ruby On Rails foi escolhido porque fornece recursos necessários para o desenvolvimento da área administrativa e da API que é utilizada pelos aplicativos.

2.2.8 POSTGRESQL

PostgreSQL é um Sistema de Gerenciamento de Banco de Dados (SGDB) SQL de código aberto com mais de 15 anos de desenvolvimento ativo e com uma arquitetura com forte reputação de confiabilidade, integridade de dados e precisão. Funciona nos principais sistemas operacionais como Linux, UNIX e Windows e possui interfaces de programação nativas para Ruby, C/C++, Java, Perl, entre outras. (GROUP, 2016).

2.2.9 SCRUM

A metodologia de desenvolvimento ágil Scrum foi utilizada no desenvolvimento dos aplicativos e também do sistema. Scrum é um *framework* para desenvolver e manter produtos complexos, utilizado desde a década de 90 no qual é possível empregar vários processos ou técnicas para o desenvolvimento de software. (SCHWABER; SUTHERLAND, 2011).

No Scrum, os projetos são divididos em ciclos chamados de *Sprints*, sendo que cada ciclo representa um período de tempo no qual um conjunto de atividades deve ser executado. As funcionalidades que serão implementadas em um projeto são mantidas em uma lista chamada de Product Backlog. No início de cada sprint ocorre uma reunião de planejamento conhecida como Sprint Planning Meeting onde o Product Owner prioriza os itens do Product Backlog e a equipe de desenvolvimento seleciona as atividades que serão desenvolvidas neste Sprint. As tarefas selecionadas para o Sprint são transferidas do Product Backlog para o Sprint Backlog (DESENVOLVIMENTOÁGIL.COM.BR, 2011).

Diariamente em um Sprint é realizada uma reunião com objetivo de conversar sobre o que foi desenvolvido no dia anterior, identificar impedimentos e priorizar o trabalho do dia. Ao final de cada Sprint a equipe realiza uma Sprint Review Meeting, que é uma reunião onde são apresentadas as funcionalidades implementadas. Também é realizada a Sprint Retrospective, e é iniciado o planejamento da próxima Sprint (DESENVOLVIMENTOÁGIL.COM.BR, 2011).

A utilização do Scrum será importante no desenvolvimento para o controle e divisão das tarefas ao longo do tempo.

3 METODOLOGIA

Os passos metodológicos estabelecidos para a realização deste projeto são:

- **Análise do problema e definição dos requisitos:** esta fase permite que o problema e as necessidades sejam melhor compreendidas, possibilitando a definição dos requisitos do sistema a ser desenvolvido. Basicamente, as tarefas realizadas nesta fase consistem em:
 - Análise e estudo do estacionamento regulamentado em vias públicas da cidade de Guarapuava-PR;
 - Análise e estudo de outros projetos do estado da arte;
 - Pesquisa utilizando formulário para avaliar o interesse dos motoristas em usar uma nova ferramenta para pagamento de estacionamentos regulamentados.
- **Modelagem e diagramação do sistema:** esta fase permite arquitetar a estrutura do sistema por meio de recursos de modelagem, além de permitir uma análise mais aprofundada sobre a integração das tecnologias utilizadas. Basicamente, esta fase será composta pelas seguintes atividades:
 - Modelagem de dados;
 - Prototipagem das telas.
- **Desenvolvimento do sistema web referente à área administrativa:** esta fase consiste no desenvolvimento do sistema de gerenciamento de estacionamentos regulamentados e do site para a administração do sistema.
- **Desenvolvimento do sistema móvel referente à área dos usuários do estacionamento regulamentado:** esta fase consiste no desenvolvimento de uma aplicação web e aplicativo móvel para uso dos usuários do estacionamento regulamentado.

- **Desenvolvimento do sistema móvel referente à área dos fiscais do estacionamento regulamentado:** esta fase consiste no desenvolvimento de um aplicativo móvel para uso dos fiscais do estacionamento regulamentado.
- **Validação e Verificação dos requisitos:** esta fase consiste na validação dos requisitos a fim de saber se todos os requisitos foram implementados corretamente e verificação junto aos interessados pelo aplicativo a fim de conhecer o seu grau de satisfação com o sistema desenvolvido.

4 ANÁLISE E PROJETO

Neste capítulo serão apresentados os resultados das atividades de Análise e Projeto, onde os parâmetros necessários para o desenvolvimento da área administrativa e das aplicações móveis.

4.1 ANÁLISE DO SISTEMA

Como resultado da análise do problema e definição dos requisitos, as Histórias (do Scrum) e a modelagem de dados foram desenvolvidas. Também foi realizada uma pesquisa com possíveis usuários do WayPark.

4.1.1 HISTÓRIAS

O Product Backlog do Scrum foi criado como resultado da análise do problema e definição dos requisitos:

- Como usuário desejo realizar um cadastro para ter acesso ao aplicativo;
- Como usuário desejo poder ativar cartões/tíquetes de estacionamento através do aplicativo para smartphone;
- Como usuário desejo ter controle de meus créditos e acesso ao histórico de ativações;
- Como fiscal desejo poder verificar se um carro está em situação regular a partir da placa utilizando o aplicativo para fiscais;
- Como fiscal desejo poder lançar multas a um veículo irregular a partir da placa utilizando o aplicativo para fiscais;
- Como administrador desejo poder ter controle total do sistema;

4.1.2 PESQUISA

Foi realizada uma pesquisa utilizando um formulário que foi distribuído na Internet através de redes sociais entre os dias 20 e 24 de novembro de 2015 para moradores de Guarapuava-PR e acadêmicos e servidores da UTFPR - Câmpus Guarapuava. Esta pesquisa foi respondida por 46 pessoas e coletou dados referentes às dificuldades de uso do estacionamento regulamentado na cidade de Guarapuava-PR e interesse dos motoristas em utilizar uma nova ferramenta de pagamento nos estacionamentos.

Entre os resultados da pesquisa, constatou-se que:

- 89% dos pesquisados utilizam automóvel para se locomover pela cidade;
- 100% encontram dificuldades para estacionar no centro da cidade, sendo que desses, 80% encontram dificuldades muitas vezes e 20% encontram dificuldades algumas vezes;
- Numa escala de 1 a 5, sendo 1 como nada satisfeito e 5 como muito satisfeito, nenhum dos pesquisados avalia a forma de cobrança atual com 5 pontos, enquanto 9% avaliam com 4 pontos, 40% com 3 pontos, 31% com 2 pontos e 20% com 1 ponto;
- 13% já receberam pontos na carteira por não saber que recebeu multa pelo agente do estacionamento;
- 69% mostraram interesse em utilizar um aplicativo para smartphone para pagar pelo estacionamento regulamentado;
- 84% dos pesquisados preferem que a cobrança seja proporcional aos minutos estacionados, enquanto 16% preferem a cobrança por cartões de 1 ou 2 horas;
- 75% consideram a compra online a melhor forma para compra de créditos para estacionamento;

A pesquisa completa e seus resultados estão listados no Apêndice A.

4.2 PROJETO DO SISTEMA

O projeto do sistema foi construído com base nos requisitos e parâmetros definidos na análise. Nesta seção serão apresentados o estado atual de desenvolvimento e as atividades que serão futuramente desenvolvidas.

4.2.1 CAIXA DO PRODUTO

A Caixa do Produto é uma técnica de análise para entender melhor os requisitos levantados. Tem o objetivo de criar a visão do produto como se fosse vendido em uma caixa numa prateleira, estimulando a equipe de projeto a criar uma imagem que represente o mesmo. Além disso, esta técnica é importante para compartilhar uma mesma visão do produto entre os envolvidos, incluindo os clientes.

Ao desenhar a caixa do produto, deve-se trabalhar um visual juntamente com informações que chamem a atenção do usuário, exibindo as principais funcionalidades da aplicação. Enquanto a frente da caixa destaca o nome e imagem ou logotipo do produto, a parte de trás elenca as funcionalidades e outras informações.

A Figura 3 exibe o modelo da caixa do produto na visão frontal. O logotipo foi destacado, assim como o nome e o slogan. Logo abaixo, foram elencadas algumas funcionalidades e vantagens do produto.



Figura 3: Caixa do Produto - Visões frontal e traseira

4.2.2 MODELAGEM DE DADOS

A Figura 4 exibe a modelagem de dados que apresenta a estrutura que será utilizada na área do servidor e administrativa do sistema. Ela mostra a relação entre clientes e cartões/tíquetes expressa no banco de dados, além das multas, agentes de estacionamento, administradores e configurações do sistema. Esta modelagem foi resultado do levantamento de requisitos realizada na fase de análise e foi construída utilizando MySQL Workbench¹.

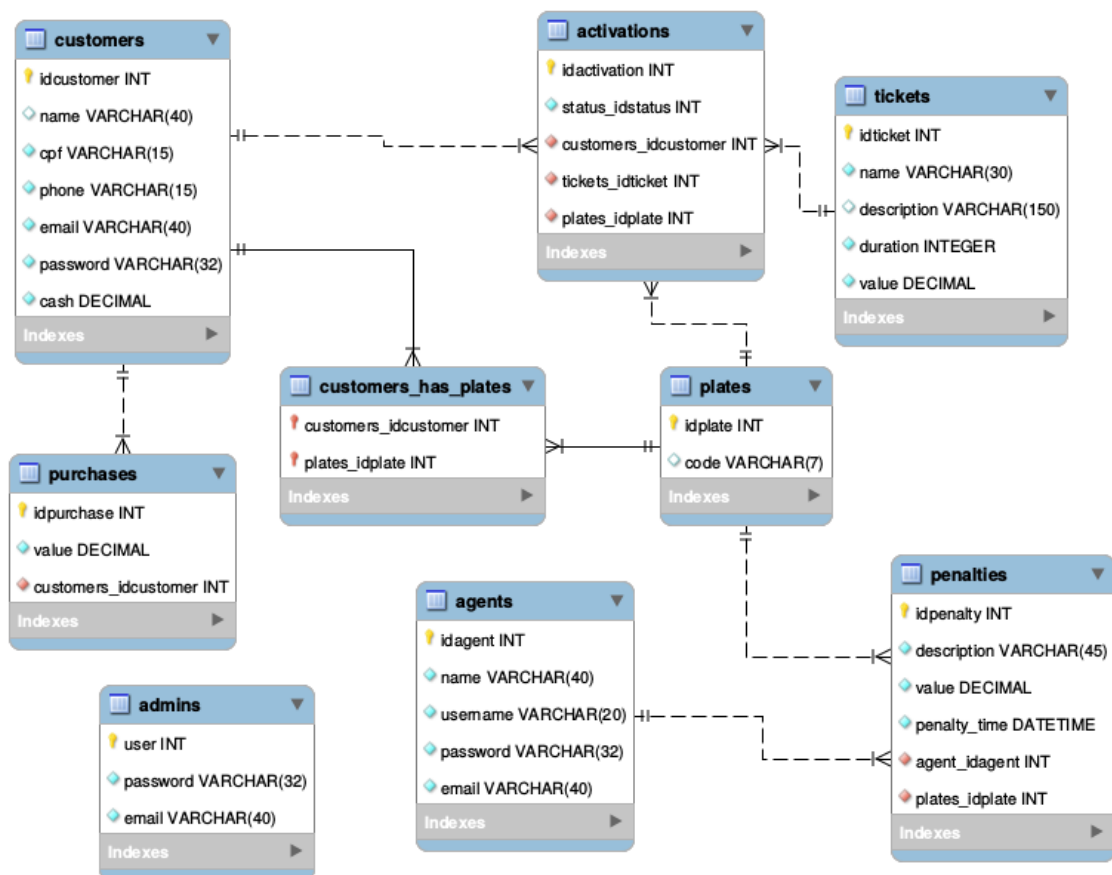


Figura 4: Modelagem de dados

4.2.3 PROTÓTIPOS DAS TELAS

Foram desenvolvidos os *wireframes* de algumas das telas do aplicativo para os usuários do estacionamento. Eles serão utilizados como base para a construção do layout e disposição dos elementos na tela. Os wireframes foram desenvolvidos utilizando a ferramenta Fluid UI².

¹MySQL Workbench é uma ferramenta visual unificada para a modelagem e construção do SQL do banco de dados (MYSQL, 2015)

²Fluid UI é uma ferramenta online para a construção de protótipos de aplicativos para dispositivos móveis (FLUIDUI, 2015)

A Figura 5(a) exibe o wireframe da tela principal do aplicativo, onde será possível ativar um cartão/ticket de estacionamento. Nesta tela será possível informar a placa do veículo, o tipo de ticket, e também renovar automaticamente ao exceder o tempo do ticket (quando permitido nas configurações). As informações da cidade e do saldo atual também serão visíveis.

A Figura 5(b) exibe o wireframe da tela principal do aplicativo com um ticket de estacionamento ativado. O tempo restante do ticket será exibido, além da opção de desativa-lo antes do final do tempo.

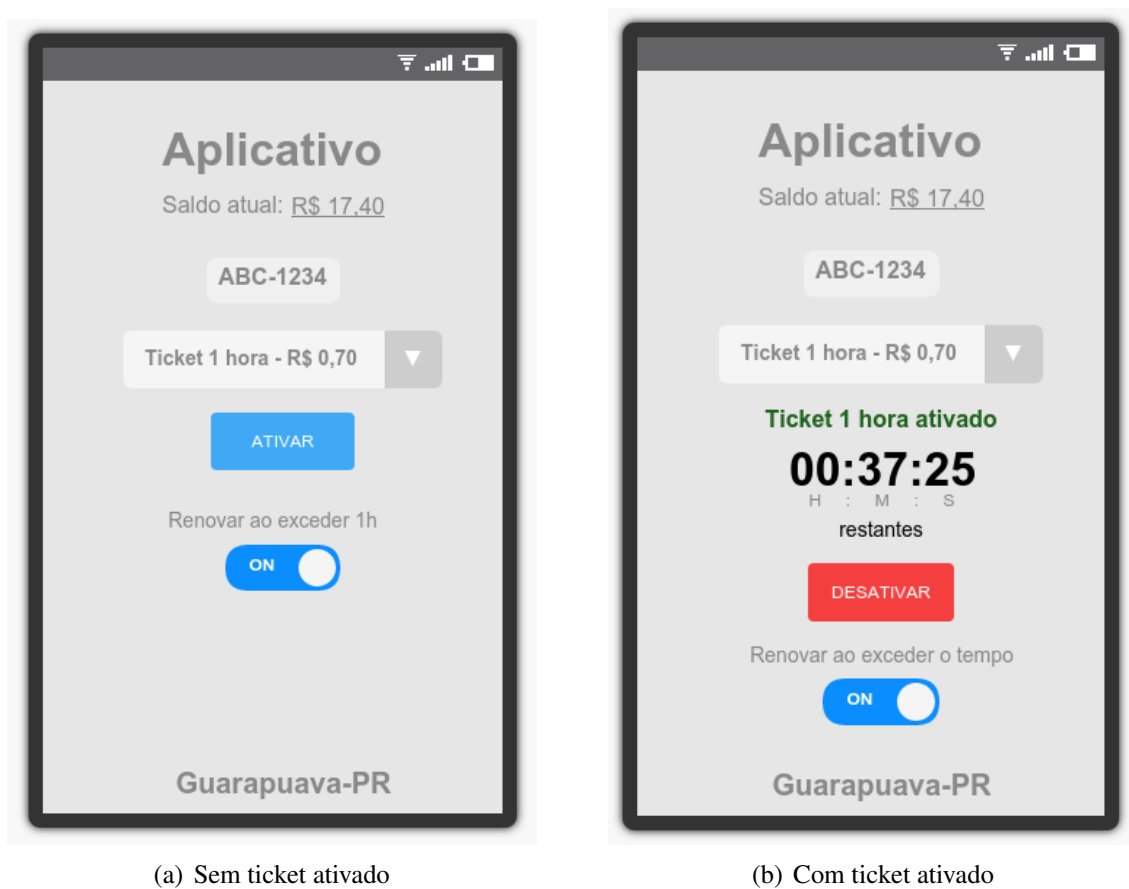


Figura 5: Wireframe da Página Principal

4.3 CENÁRIOS

A seguir serão exemplificadas algumas situações de uso do sistema, seja através dos aplicativos ou da área administrativa. Os cenários exibem situações possíveis na utilização do sistema.

Cenário 1: Uso do aplicativo dos clientes do estacionamento

- O usuário realiza seu cadastro no aplicativo;
- O servidor confirma o cadastro e autentica o usuário;
- O usuário compra créditos;
- O servidor valida esta compra enviando e solicitando informações do método de pagamento;
- O usuário digita a placa do carro e escolhe o tempo desejado na página principal;
- O usuário clica em ativar;
- O servidor valida esta ativação e retorna a confirmação para o usuário.

Cenário 2: Uso do aplicativo dos agentes do estacionamento e regularização de multas pelo usuário

- O agente de estacionamento realiza login no aplicativo;
- O servidor confirma e autentica o usuário;
- O agente de estacionamento digita uma placa e confirma;
- O servidor retorna informações sobre a solicitação;
- Com as informações do veículo, o agente pode confirmar se a situação está regular ou também pode lançar uma multa;
- O usuário visualizará esta multa pelo aplicativo;
- O usuário poderá regularizar esta multa pelo aplicativo ou diretamente no órgão responsável pelo estacionamento.

4.4 CONCLUSÃO

Os resultados obtidos nas fases de Análise e Projeto foram importantes para definir os requisitos do sistema. O êxito nesta fase permitiu o melhor aproveitamento da fase de desenvolvimento do sistema.

Enquanto a pesquisa colaborou na definição das necessidades do sistema, recebendo opinião de possíveis usuários, os cenários criados foram a visualização prévia do funcionamento do sistema com estas necessidades supridas e implementadas.

5 DESENVOLVIMENTO

O desenvolvimento do sistema foi a fase que mais concentrou os esforços dedicados ao trabalho. A divisão do sistema se dá em três partes: Área Administrativa e API, Aplicação dos Clientes e Aplicação dos Agentes de Trânsito.

Cada área possui finalidades específicas, porém o sistema necessita destas três áreas para cumprir seu propósito. Os detalhes sobre cada uma serão descritos a seguir.

5.1 ÁREA ADMINISTRATIVA E API

A área administrativa é uma aplicação construída na linguagem Ruby com o framework Ruby on Rails que também inclui a API REST.

Utilizando a arquitetura MVC (Modelo-Visão-Controlador) e prezando pela qualidade do código, o Ruby on Rails forneceu uma plataforma robusta que permitiu com que todas as funcionalidades planejadas para a área administrativa fossem desenvolvidas. Integram também esta aplicação o uso de Haml (HTML Abstraction Markup Language) que fornece uma nova sintaxe para a escrita do HTML e CORS (Cross-Origin Resource Sharing) para permitir o acesso da API pelas outras aplicações.

Foram construídos diversos testes para assegurar a integridade e funcionamento da aplicação e os dados são armazenados em um banco de dados relacional utilizando o SGBD PostgreSQL. Este banco de dados foi construído com base na modelagem desenvolvida na fase de Análise e Projeto, mas sofreu alterações para se adequar melhor ao Ruby on Rails.

5.1.1 ÁREA ADMINISTRATIVA

A Área Administrativa pode ser acessada apenas pelos administradores e oferece um controle total do sistema. Foi implementado o CRUD (Create, Read, Update and Delete) para os modelos:

- Ticket;
- Cliente;
- Ativação;
- Placa;
- Multa;
- Agente de Trânsito;
- Transação.

Embora não seja função do administrador criar ativações ou cadastrar clientes por exemplo, estas funções estão disponíveis no sistema para fins de correção ou suporte a usuários. Além disso, estes CRUDs serviram como base para implementar a API, que também utiliza dados destes modelos.

A interface gráfica foi construída utilizando o framework Materialize, que permite a criação de templates no conceito Material Design. O layout é responsivo, otimizando a navegação de dispositivos com diferentes tamanhos de tela.

A Figura 6 exibe a tela principal do do sistema, onde estão os elementos:

- Barra de navegação no topo;
- Menu principal à esquerda;
- Tickets ativos ao centro.

Pela Barra de navegação no topo é possível acessar a página de alteração de dados e também realizar o *logoff*. Pelo Menu Principal é possível acessar as diversas áreas da aplicação, sendo elas: Home (Página Principal), Tickets, Clientes, Ativações, Placas, Multas, Compras e Agentes de Trânsito. A área em que o usuário está no momento fica destacada no menu.

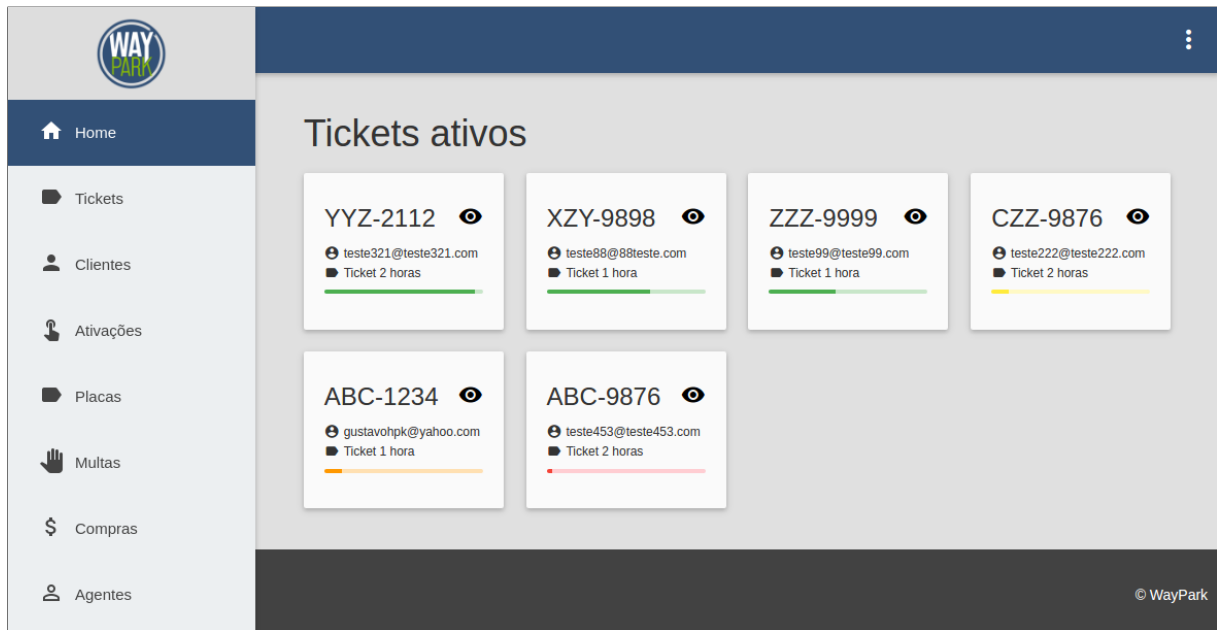


Figura 6: Página Principal

Ao centro estão listados os tickets ativos (ativações) no formato de cartões. Cada cartão exibe as seguintes informações:

- Número da Placa;
- Botão com o ícone de um olho, pelo qual é possível ver mais detalhes sobre a ativação;
- Endereço de e-mail do cliente que ativou o ticket;
- Tipo do ticket;
- Barra de progresso

A barra de progresso exibe a porcentagem de tempo passado desde que o ticket foi ativado em relação ao tempo total do ticket. Além disso, ela pode estar colorida de diferentes cores, seguindo as regras:

- Verde, quando faltam mais de 15 minutos para o tempo limite expirar;
- Amarelo, quando falta de 11 a 15 minutos para o tempo limite expirar;
- Laranja, quando faltam de 6 a 10 minutos para o tempo limite expirar;
- Vermelho, quando faltam menos de 5 minutos para o tempo limite expirar.

5.1.2 API

A API fornece as operações e dados que podem ser solicitados pelas aplicações do cliente e do agente de trânsito, mas é preciso estar autenticado nestas aplicações para poder utilizar a API. Esta autenticação ocorre através de *tokens*, os quais são criados durante o cadastro ou login dos clientes ou agentes de trânsito, e são únicos para cada conta. Este token, após confirmada a autenticação é retornado à estas aplicações, as quais armazenam e enviam-o novamente ao servidor cada vez que a API é solicitada.

Através de um *Serializer* construído para cada modelo, a API consegue listar os dados solicitados no formato JSON (Javascript Object Notation) e enviar para as aplicações. Os dados recebidos pela API também precisam estar no formato JSON. A Figura 7 exhibe a arquitetura comportamental da API.

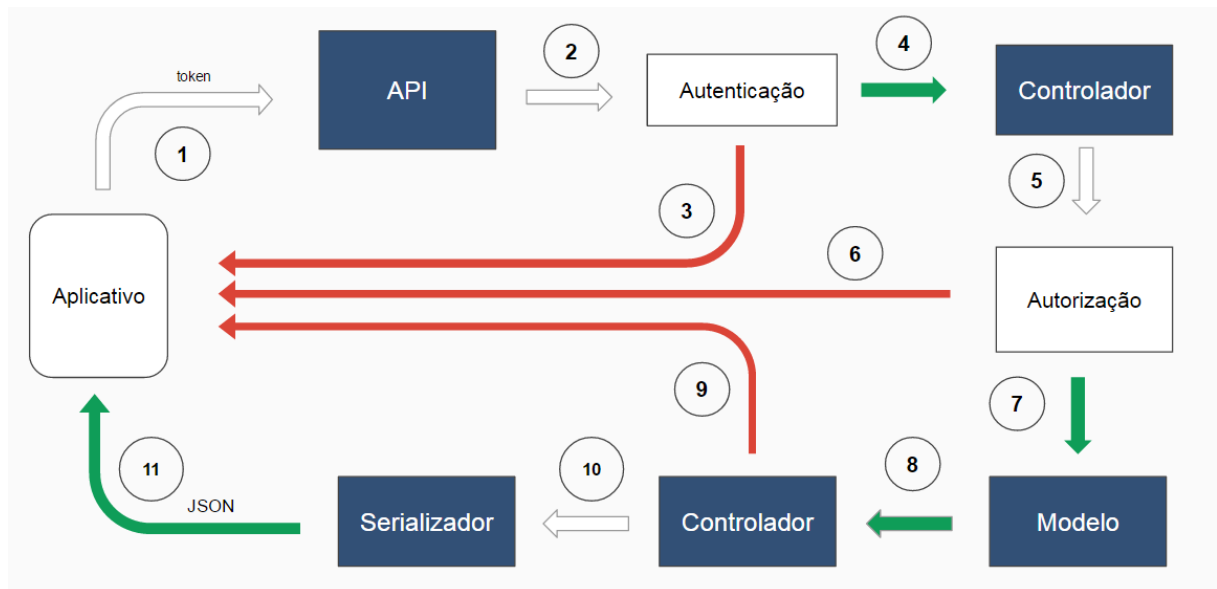


Figura 7: Arquitetura Comportamental da API

Os passos exibidos na Figura 7 são:

- 1. O aplicativo dos clientes ou dos agentes de trânsito faz uma solicitação à API, enviando o token de acesso;
- 2. A API realiza a autenticação do usuário;
- 3. Caso o token seja inválido, uma mensagem de erro é retornada;
- 4. Caso a autenticação seja confirmada, a API encaminha a solicitação ao respectivo controlador;

- 5. A API realiza a autorização, para verificar se o usuário autenticado pode realizar a solicitação;
- 6. Caso o usuário não esteja autorizado, uma mensagem de erro é retornada;
- 7. Caso o usuário esteja autorizado, o controlador solicita os dados necessários ao modelo;
- 8. Os dados são retornados ao controlador;
- 9. Caso algum erro tiver ocorrido durante o processo, uma mensagem de erro é retornada;
- 10. Caso contrário, os dados são encaminhados ao respectivo serializador;
- 11. Os dados são retornados no formato JSON para o aplicativo.

5.2 APLICAÇÃO DOS CLIENTES

A aplicação dos clientes foi construída com o framework Ionic 2 utilizando as tecnologias HTML 5, CSS 3, SASS, Typescript e Javascript. Esta aplicação fornece os recursos para que o usuário utilize as funcionalidades do WayPark.

Por ser construída com o Ionic 2, a interface gráfica da aplicação possui elementos do Material Design, preza pela simplicidade e facilidade de uso. Uma barra de navegação no topo, um menu lateral e um rodapé com guias fornecem ao usuário caminhos para acessar todas as funcionalidades do aplicativo dispostas em diferentes páginas.

A comunicação com a área administrativa através da API se dá pela comunicação de dados no formato JSON. Nenhum dado é salvo na aplicação, exceto aqueles necessários para o funcionamento da mesma, como token de autenticação.

O token de autenticação é um código único gerado durante o cadastro e recebido no momento do login, ficando armazenado no dispositivo do cliente utilizando LocalStorage. Além disso, o token é enviado para a API a cada requisição feita à mesma, e a aplicação do servidor verifica se a operação solicitada é permitida para o cliente que a solicita, impedindo assim a visualização de dados de outros clientes.

A Figura 8 exhibe a arquitetura comportamental da Aplicação dos Clientes durante a ação de ativação de ticket.

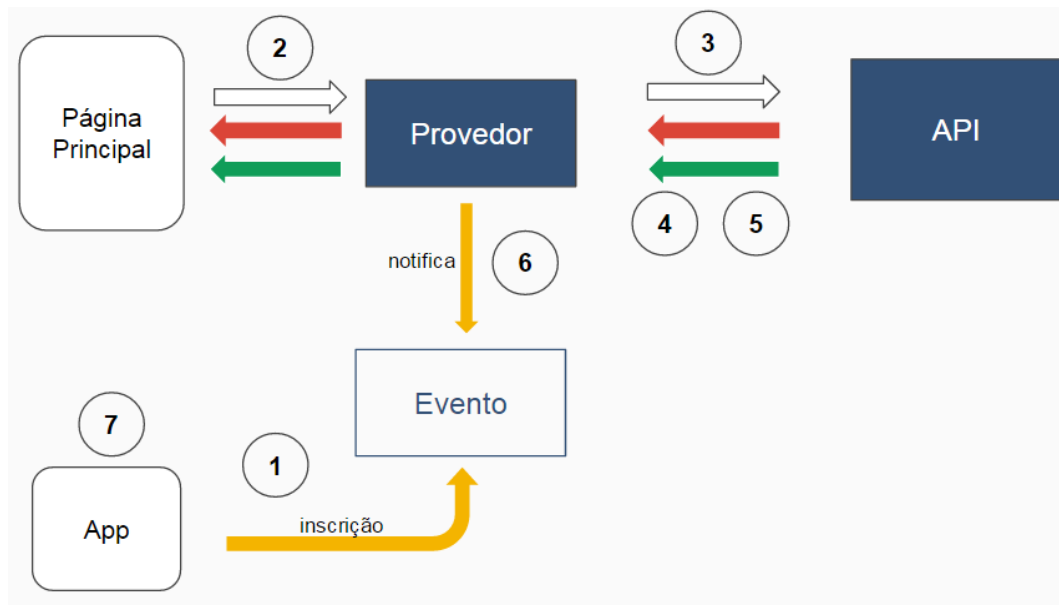


Figura 8: Arquitetura Comportamental da Aplicação dos Clientes

Os passos exibidos na Figura 8 são:

- 1. Ao ser inicializado, o aplicativo se inscreve em um evento para verificar alterações de dados do cliente;
- 2. A página principal envia os dados da ativação de ticket para o provedor;
- 3. O provedor envia uma solicitação à API, encaminhando no formato JSON os dados recebidos;
- 4. Caso tenha ocorrido um erro na ativação de ticket, a API retorna uma mensagem de erro ao provedor, que por sua vez retorna este erro à página principal;
- 5. Caso contrário, o provedor retorna a mensagem de sucesso à página principal;
- 6. O provedor notifica o evento que verifica alterações de dados do cliente que os dados foram alterados (neste exemplo, o saldo foi debitado);
- 7. Após receber a notificação pelo evento, o aplicativo atualiza o saldo do cliente no menu e no LocalStorage.

A Figura 9 exibe a tela de login do aplicativo. Ela é utilizada para a autenticação, obrigatória para utilizar o aplicativo, e para isso, é necessário informar o endereço de e-mail e senha da conta do usuário. Caso o usuário não possua uma conta, pode acessar a página de cadastro através do botão "Registrar".

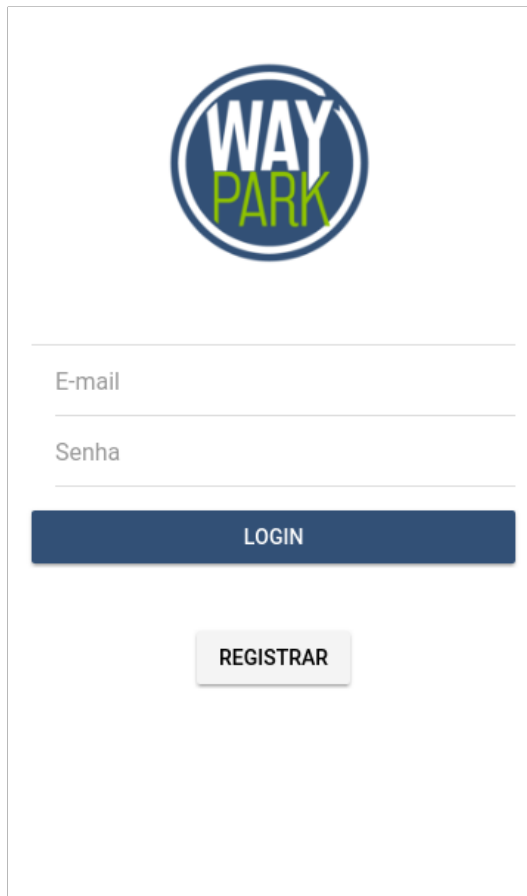
A imagem mostra a interface de login do aplicativo Way Park. No topo, há o logotipo circular "WAY PARK" em azul e verde. Abaixo, há dois campos de entrada de texto: "E-mail" e "Senha". Abaixo dos campos, há um botão azul escuro com o texto "LOGIN" em branco. Abaixo do botão "LOGIN", há um botão cinza com o texto "REGISTRAR" em preto.

Figura 9: Página de Login

A Figura 10(a) exibe a tela de cadastro, necessário para autenticar-se. O usuário precisa informar seu nome, telefone, endereço de e-mail e senha (e repetir a senha no campo de confirmação).

Estes dados serão enviados para o servidor, que validará as informações, verificando se o endereço de e-mail já não está cadastrado. Se os dados passarem pela validação, o cadastro é realizado e o usuário é automaticamente autenticado, sendo redirecionado para a página principal.

Se a validação retornar erros, eles serão exibidos na cor vermelha abaixo do formulário, como mostra a Figura 10(b). Esta forma de validação e exibição de erros é a mesma utilizada em outros formulários da aplicação.

Registrar

Nome

Telefone

E-mail

Senha

Repetir senha

ENVIAR VOLTAR

Nome

Telefone

E-mail

Senha

Repetir senha

E-mail não pode ficar em branco

Senha não pode ficar em branco

Nome não pode ficar em branco

ENVIAR VOLTAR

(a) Formulário de Cadastro

(b) Formulário com erros

Figura 10: Página de Cadastro

A Figura 11(a) exibe a tela principal do aplicativo, onde estão os elementos:

- Barra de Navegação no topo
- Título da página e saldo do usuário abaixo da Barra de Navegação
- Formulário de ativação de ticket no centro
- Botão "Menu" no canto superior esquerdo, pelo qual é possível acessar o menu do aplicativo
- Abas na parte inferior do aplicativo, onde é possível alternar entre as páginas Principal (Home), Tickets, Multas e Compras.

O formulário de ativação permite que o usuário ative um ticket, selecionando a placa e o ticket. Caso nenhuma placa esteja cadastrada, o aplicativo informará o usuário e exibirá o formulário para cadastro de placa. Caso o usuário não possua créditos suficientes na conta, será necessário realizar a compra de créditos. Também há na mesma página os botões para cadastro de placa e compra de créditos.

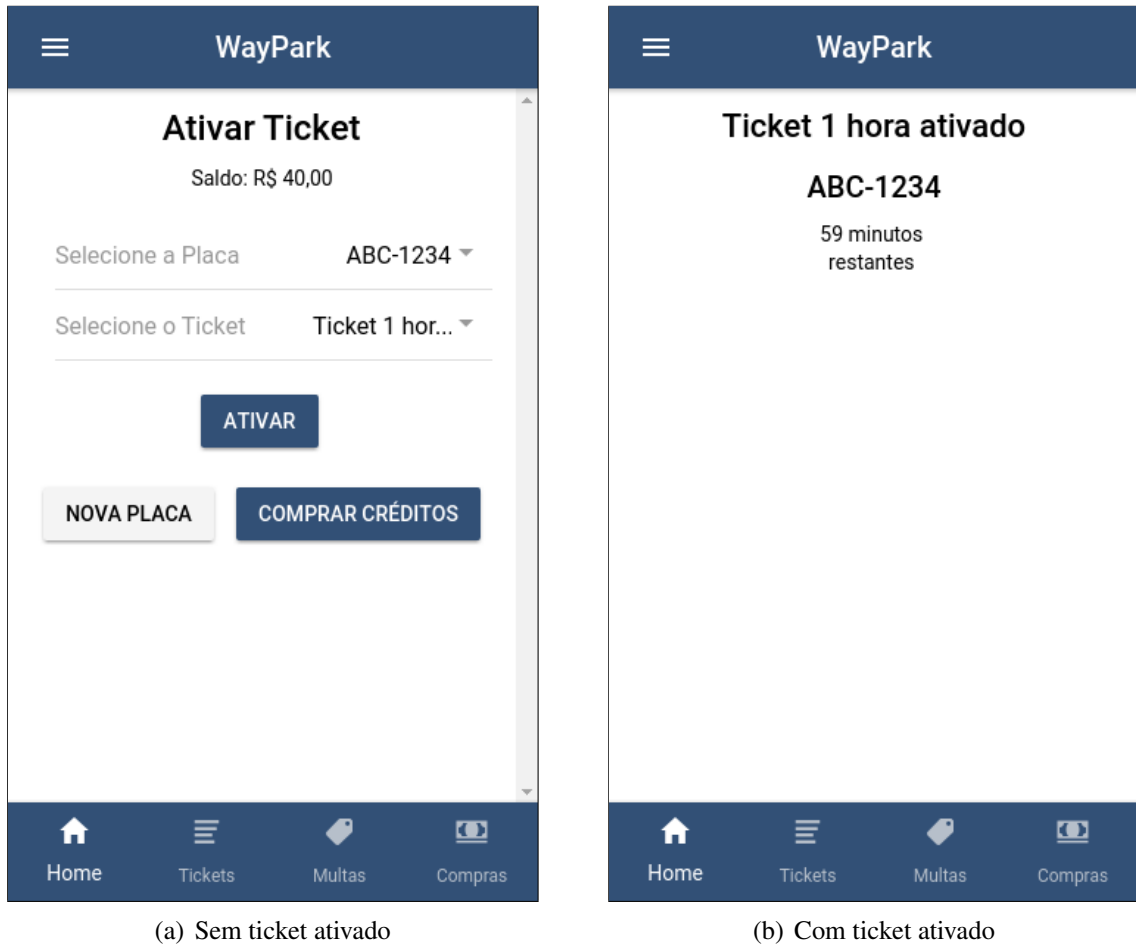


Figura 11: Página Principal

Caso o dispositivo do usuário não esteja conectado à internet ou tenha ocorrido algum problema com a conexão, o botão de ativação de ticket é desabilitado e o aviso "Conexão perdida com o servidor" é exibido. A aplicação tentará reestabelecer a conexão a cada 30 segundos, mas o usuário também pode clicar no botão "Tentar Novamente" para tentar esta reconexão.

Caso um ticket esteja ativo, a página principal exibirá o tempo restante da validade deste ticket. Também há um botão para desativar este ticket, como mostrado na Figura 11(b).

A Figura 12(a) exibe o menu principal do aplicativo, pelo qual é possível acessar as páginas "Comprar Créditos", "Sobre", "Contato", "Minha Conta" e também realizar o logoff.

A Figura 12(b) exibe a página "Minha Conta". Esta página possui um formulário pelo qual é possível alterar informações como nome, telefone e senha. Ao clicar no botão "Salvar", as informações são enviadas para o servidor e um aviso de sucesso ou erro é retornado.

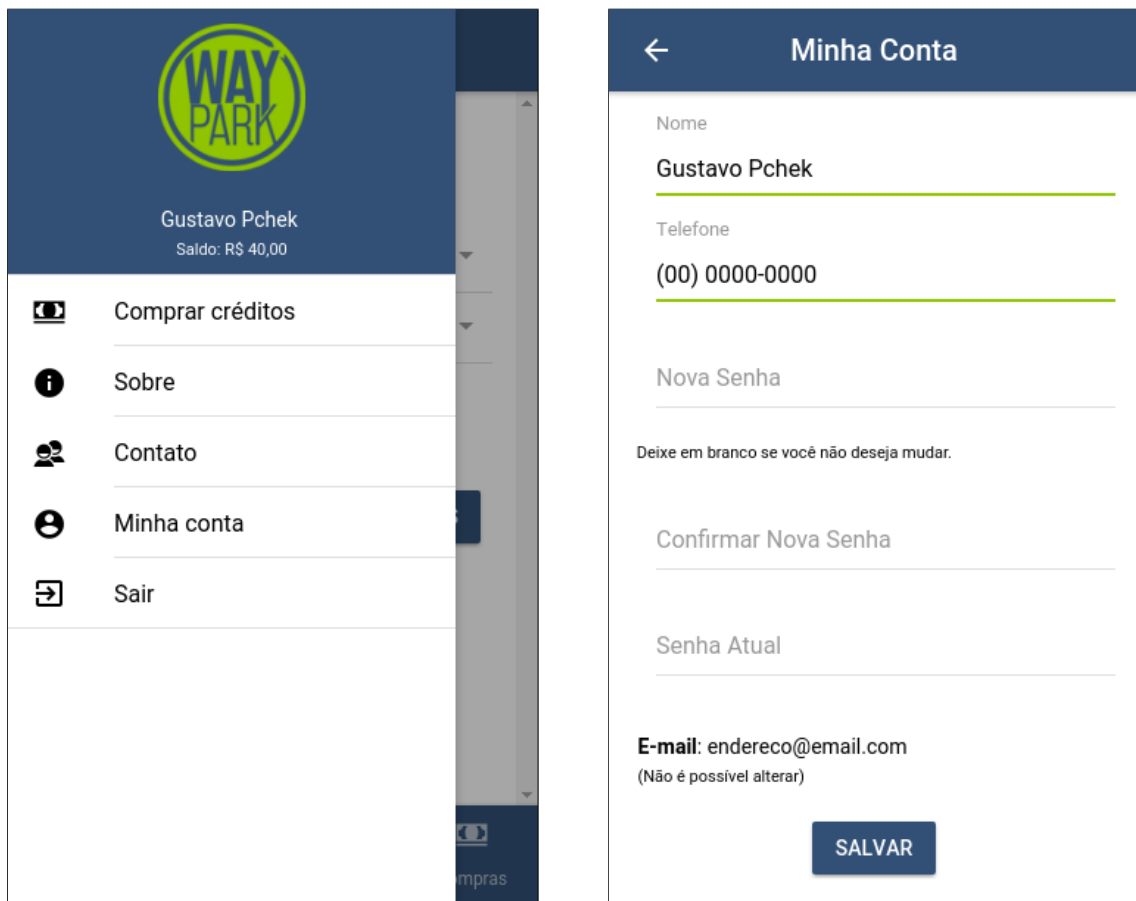
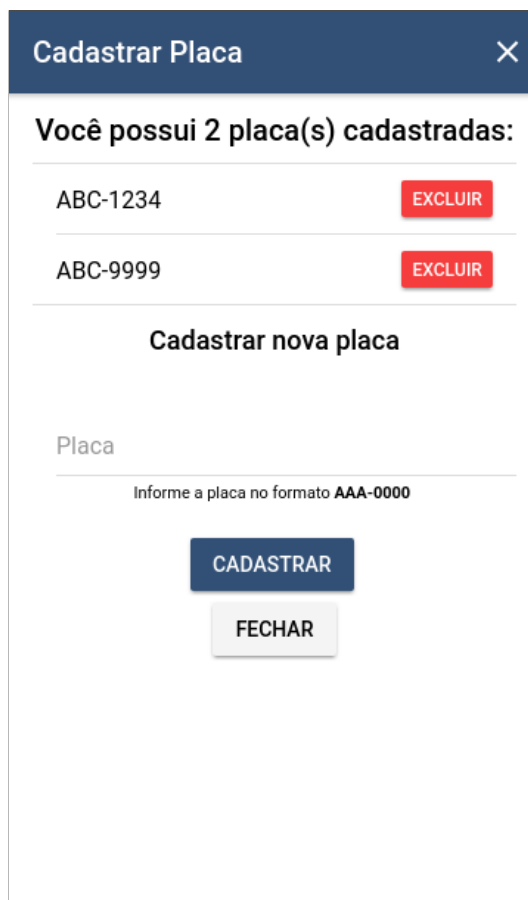


Figura 12: Menu e Página Minha Conta

A Figura 13 exibe o *modal* para cadastro e exclusão de placas. Ao preencher o campo e clicar em "Cadastrar", uma solicitação é enviada ao servidor, que cadastra a placa no sistema e vincula-a ao usuário.

Ao clicar em algum botão "Excluir", um alerta de confirmação é exibido, para que o usuário possa confirmar ou cancelar a exclusão da placa. Caso confirme, uma solicitação é enviada ao servidor, mas a placa não é excluída do banco de dados para manter o histórico, mas sim desvinculada do usuário.



Cadastrar Placa ✕

Você possui 2 placa(s) cadastradas:

ABC-1234	EXCLUIR
ABC-9999	EXCLUIR

Cadastrar nova placa

Placa

Informe a placa no formato **AAA-0000**

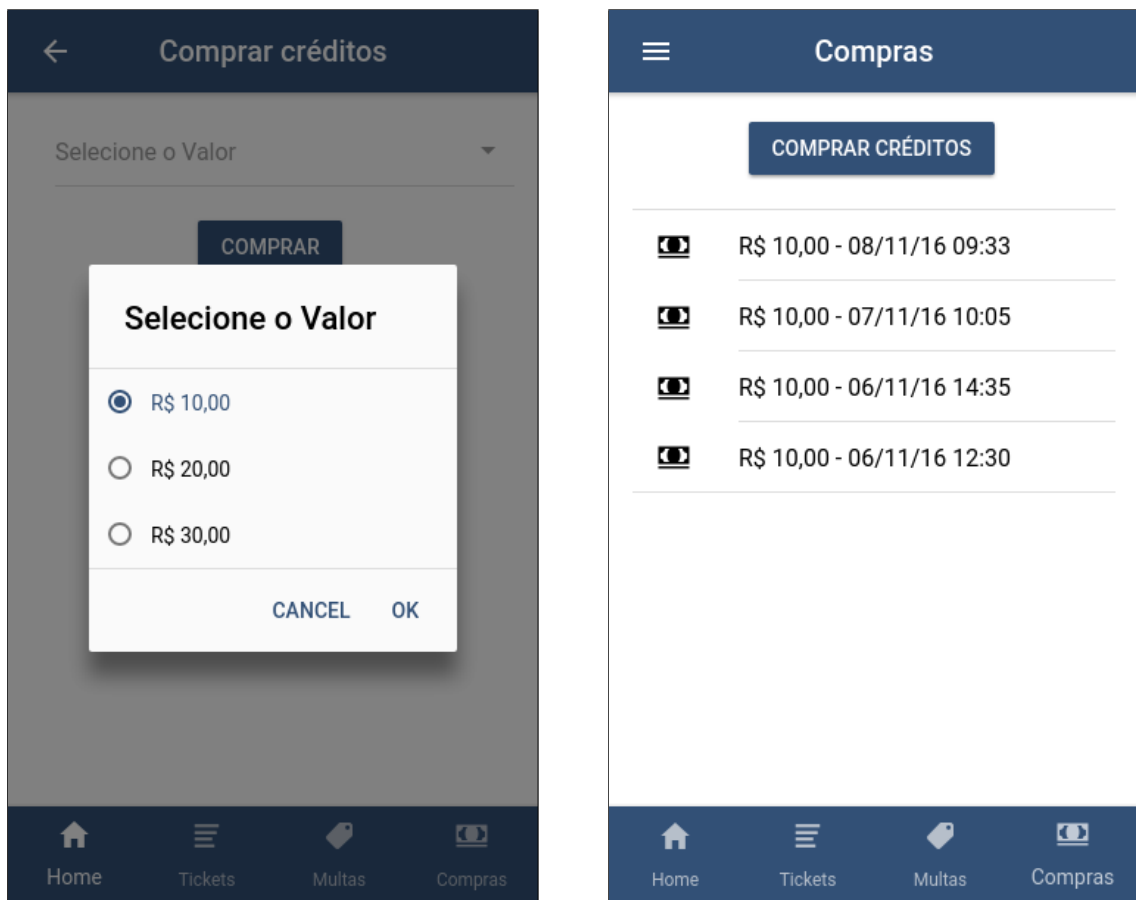
CADASTRAR

FECHAR

Figura 13: Modal Placa

A Figura 14(a) exibe a página para compra de créditos. O usuário seleciona um dos valores predefinidos através do campo "Valor" e clica no botão "Comprar". Um alerta de confirmação é exibido ao usuário, que pode confirmar ou cancelar a compra. Caso a compra ocorra com sucesso, o usuário é direcionado à página principal.

A página "Compras" exibida na Figura 14(b) mostra o histórico de compra de créditos. Ao clicar em um item da lista, a página da compra é exibida, a qual contém os detalhes como valor, data e hora.



(a) Página Comprar Créditos

(b) Histórico de Compras

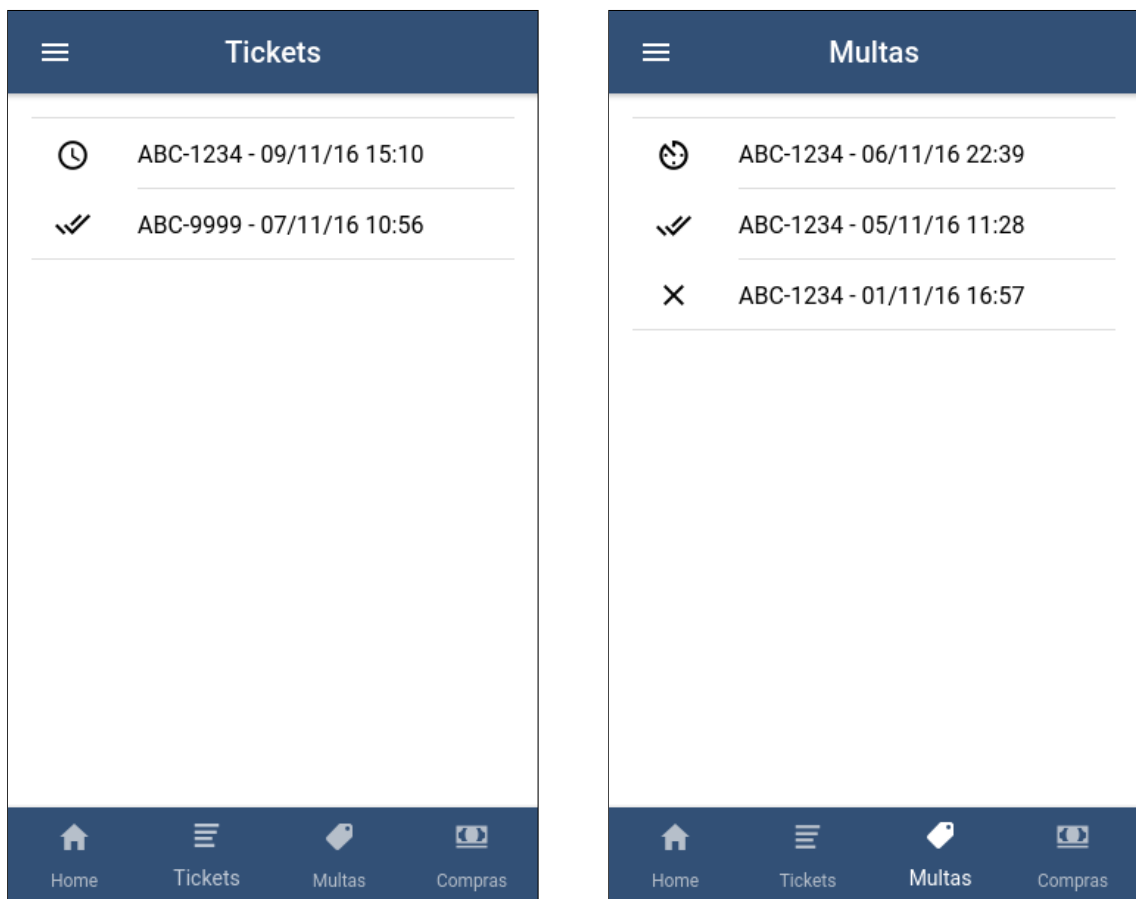
Figura 14: Páginas de Compra e Histórico de Compras

A página "Tickets" exibida na Figura 15(a) mostra o histórico dos tickets ativados pelo usuário. O ícone do relógio aparece ao lado dos tickets que estão ativos, e o ícone de confirmação aparece ao lado dos tickets cujas ativações já expiraram o tempo limite do ticket. Ao clicar em um item da lista, a página da ativação é exibida, a qual contém os detalhes como placa, tipo de ticket e preço, data e hora.

A página "Multas" exibida na Figura 15(b) mostra as multas de estacionamento que os veículos cujas placas vinculadas ao usuário receberam. Ao lado da multa, podem ser exibidos os seguintes ícones:

- Cronômetro, quando o prazo para pagamento ainda não acabou;
- "X", quando o prazo para pagamento acabou e a multa não foi paga;
- Confirmação ou "V", quando a multa já foi paga.

Ao clicar em um item da lista, a página da multa é exibida, a qual contém os detalhes como placa, data e hora, valor, data e hora limites para pagamento e estado do pagamento. Também há um botão que permite ao usuário pagar a multa pelo próprio aplicativo.



(a) Histórico de Tickets Ativados

(b) Histórico de Multas

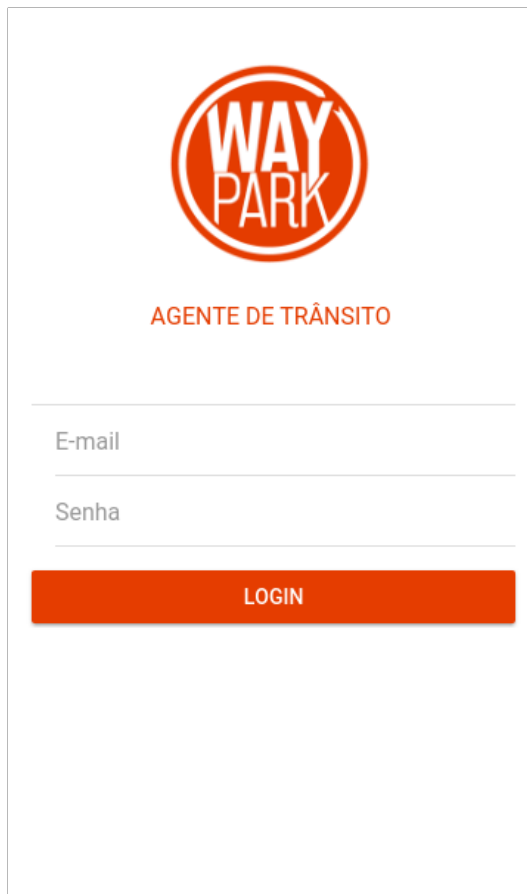
Figura 15: Históricos de Tickets Ativados e Multas

5.3 APLICAÇÃO DOS AGENTES DE TRÂNSITO

Assim como a aplicação para os clientes, a aplicação para os agentes de trânsito também foi construída utilizando o framework Ionic 2. Ambas são semelhantes em estrutura e interface gráfica, porém diferem em funcionalidades. A autenticação e acesso à API ocorre da mesma forma que na aplicação dos clientes.

A interface gráfica é semelhante à aplicação dos clientes, mas cores da aplicação dos agentes de trânsito são diferentes. A cor primária foi substituída por vermelho e a cor secundária, por amarelo.

A Figura 16 exibe a tela de login do aplicativo. Ela é utilizada para a autenticação, obrigatória para utilizar o aplicativo, e para isso, é necessário informar o endereço de e-mail e senha da conta do usuário. Diferente da aplicação dos clientes, não é possível se cadastrar pelo aplicativo. Os agentes de trânsito devem ser cadastrados pela área administrativa.



A imagem mostra a tela de login do aplicativo Way Park para agentes de trânsito. No topo, há o logotipo circular em vermelho com o texto "WAY PARK" em branco. Abaixo do logotipo, o texto "AGENTE DE TRÂNSITO" é exibido em vermelho. A tela contém dois campos de entrada de texto: "E-mail" e "Senha", ambos com linhas de base cinzas. Abaixo dos campos, há um botão retangular em vermelho com o texto "LOGIN" em branco.

Figura 16: Página de Login

A Figura 17 exibe a tela principal do aplicativo, onde estão os elementos:

- Barra de Navegação no topo
- Formulário de consulta de placa no centro
- Botão "Menu" no canto superior esquerdo, pelo qual é possível acessar o menu do aplicativo
- Abas na parte inferior do aplicativo, onde é possível alternar entre as páginas Principal (Home) e Multas.

Pelo formulário é possível consultar o estado (regular ou irregular) de um veículo. Ao informar a placa e clicar no botão "consultar", a aplicação consulta os dados do servidor pela API, que retorna se o veículo possui algum ticket ativado, além de exibir o histórico recente de ativações.

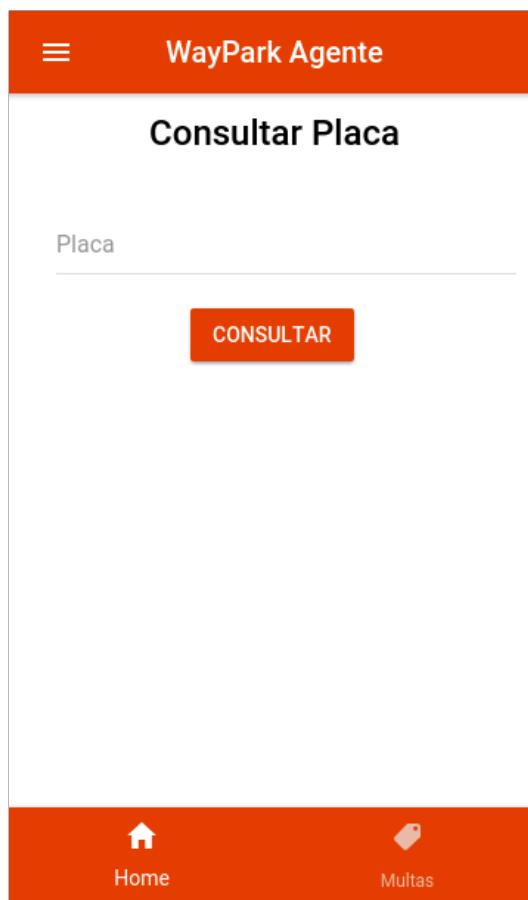


Figura 17: Página Principal

A página "Multas" exibe o histórico de multas aplicadas pelo agente de trânsito. Ao clicar em um item da lista, a página da multa é exibida, a qual contém as informações como placa, valor, data e hora e data e hora limites para pagamento.

A Figura 18 exibe o menu principal do aplicativo, pelo qual é possível acessar as páginas "Sobre", "Contato", "Minha Conta" e também realizar o logoff.

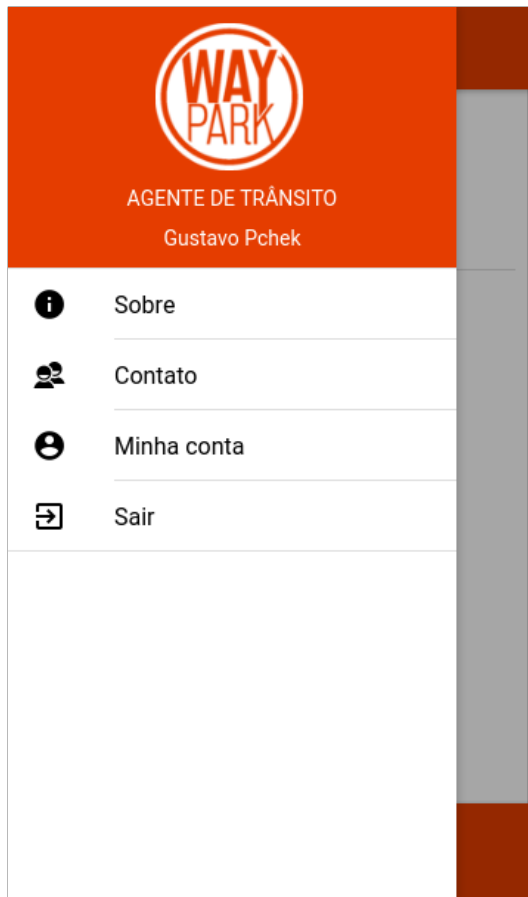


Figura 18: Menu Principal

5.4 CONCLUSÃO

O processo de desenvolvimento do trabalho ocorreu dividido em sprints, seguindo a metodologia Scrum. Cada sprint possuiu 2 semanas, e no final de cada uma era realizada uma reunião com o professor orientador, que fazia o papel do cliente, analisando e avaliando o progresso do sistema. A utilização deste processo mostrou-se eficaz, pois permitiu uma avaliação periódica das funcionalidades do sistema.

Durante o desenvolvimento foram identificadas necessidades não previstas na fase de análise e projeto, e elas foram implementadas até o final do processo. Os protótipos de telas desenvolvidos na fase de Análise e Projeto foram utilizados como base para construção do layout da aplicação dos clientes, porém sofreram mudanças e aprimoramentos para adequá-los ao Material Design e a um melhor funcionamento do sistema.

As tecnologias Angular e Ionic 2 estão em constante atualização¹². O aprendizado das mesmas pelo autor ocorreu juntamente com o início da fase de desenvolvimento, e por isso foram as que mais foram desafiadoras e ocasionaram algumas dificuldades durante o processo.

A aplicação em Ruby on Rails da Área Administrativa e API também exigiram importantes esforços principalmente na construção da API com autenticação e autorização, a qual impede que as aplicações móveis acessem informações que não podem visualizar.

A divisão do sistema em três partes - Aplicação dos Clientes, Aplicação dos Agentes de Trânsito e Área Administrativa e API - permitiu um melhor estruturamento das funcionalidades e concentrando todos os dados do sistema apenas em um local. E a comunicação entre elas através da API REST mostrou-se eficiente, atendendo às necessidades do sistema.

¹Código-fonte do Angular disponível em: <https://github.com/angular/angular>

²Código-fonte do Ionic 2 disponível em: <https://github.com/driftyco/ionic/>

6 RESULTADOS

Neste capítulo serão apresentados os resultados dos testes realizados com o protótipo do WayPark. Os testes realizados consistiam na simulação da utilização da aplicação dos clientes. Este teste foi realizado presencialmente por algumas pessoas e o restante recebeu as orientações através de ferramentas de comunicação online como Messenger e WhatsApp.

As pessoas que participaram dos testes foram informadas que o aplicativo que testariam é um protótipo e orientados a acessar um *link* disponibilizado pelo autor através do computador ou smartphone. Além disso, foram orientados a seguir os seguintes passos:

- Realizar o cadastro
- Comprar créditos
- Ativar um ticket

Os créditos adquiridos no sistema eram fictícios, não havendo a necessidade de compra real de créditos. Como o teste também consistiu em avaliar a usabilidade do sistema, não foram passados detalhes sobre como realizar estes passos. As pessoas também foram orientadas a preencher um questionário sobre as impressões do teste e opiniões sobre o aplicativo. Este formulário foi preenchido manualmente pelas pessoas que fizeram o teste presencial, e digitalmente através do acesso a um *link* enviado junto com as orientações.

No total, seis pessoas realizaram o teste da aplicação dos clientes e responderam ao questionário do formulário. Com base nas respostas, é possível afirmar que a maioria das pessoas não teve dificuldades para utilizar o aplicativo, como mostra a Figura 19.

Você encontrou dificuldades para utilizar o aplicativo? (6 respostas)

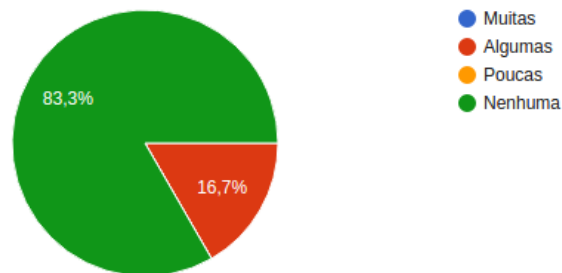


Figura 19: Respostas da pergunta: Você encontrou dificuldades para utilizar o aplicativo?

A interface gráfica foi avaliada com notas de 1 a 5, sendo 1 pior e 5 melhor. Metade dos usuários concedeu nota 5, e a outra metade notas 3 e 4, como mostra a Figura 20.

Como você avalia a Interface Gráfica, em uma nota de 1 a 5? (6 respostas)

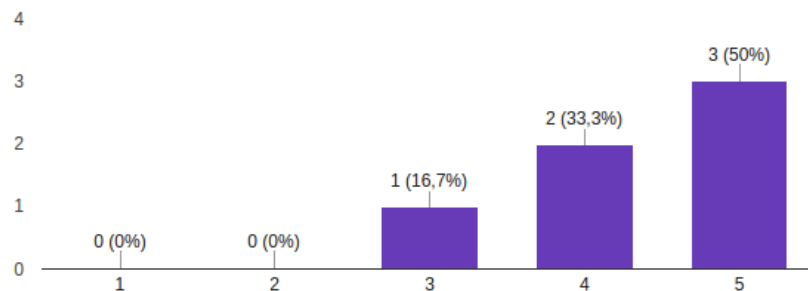


Figura 20: Respostas da pergunta: Como você avalia a Interface Gráfica, em uma nota de 1 a 5?

As funcionalidades "Ativar ticket" e "Ver tempo restante do ticket ativo" foram escolhidas como as que os usuários mais gostaram, como mostra a Figura 21.

Metade dos usuários consideraram que o uso do aplicativo desde o cadastro até a ativação de ticket foi rápido, como mostrado na Figura 22.

Por fim, a maioria dos usuários respondeu que utilizaria o WayPark para pagar pelo uso de estacionamentos, como exibido na Figura 23

Selecione as funcionalidades que você mais gostou no aplicativo (6 respostas)

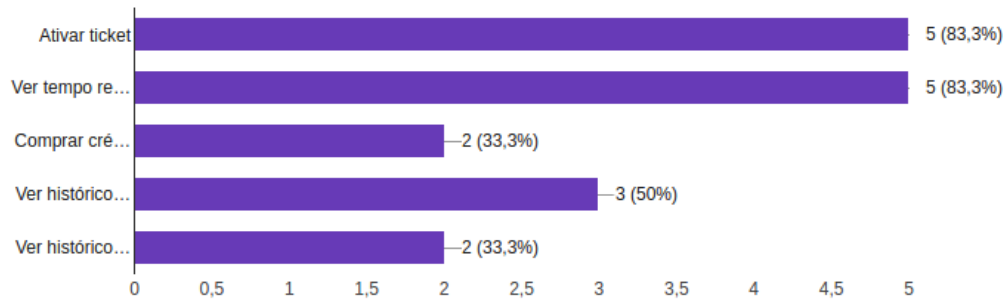


Figura 21: Respostas da questão: Selecione as funcionalidades que você mais gostou no aplicativo.

Desde o cadastro até a ativação de ticket, você acha que: (6 respostas)

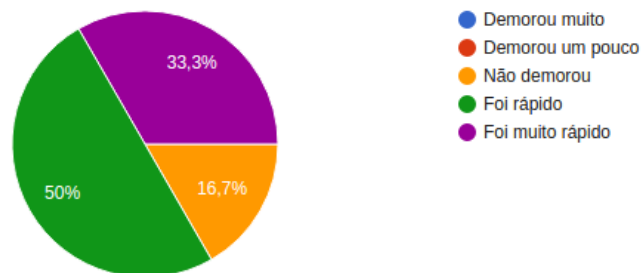


Figura 22: Respostas da questão: Deste o cadastro até a ativação do ticket, você acha que:

Você utilizaria o WayPark para pagar estacionamentos, como alternativa aos cartões de papel?

(6 respostas)

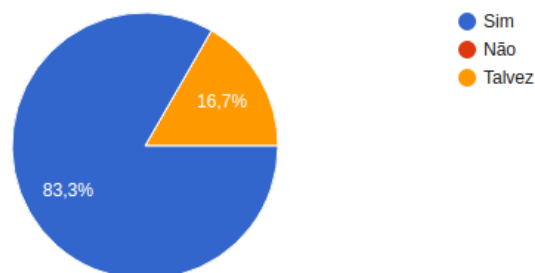


Figura 23: Respostas da pergunta: Você utilizaria o WayPark para pagar pelo uso de estacionamento, como alternativa aos cartões de papel?

7 CONSIDERAÇÕES FINAIS

O trabalho contribuirá de diversas formas na melhoria e otimização da cobrança de estacionamentos regulamentados. Espera-se que o projeto contribua positivamente em todo o processo de controle e cobrança dos estacionamentos regulamentados, seja na agilidade e melhoria do processo para o cliente do estacionamento ou na segurança e facilidade de acesso à informação pelo órgão fiscalizador.

Durante o desenvolvimento do projeto, houveram dificuldades para encontrar algumas informações sobre o funcionamento dos estacionamentos regulamentados em algumas cidades brasileiras e por isso, parte das informações foi retirada de portais de notícias na internet. Porém a pesquisa realizada com potenciais usuários do sistema contribuiu bastante para sanar parte das dúvidas existentes no início.

O conhecimento prévio adquirido das tecnologias HTML, CSS, Javascript, Ruby, Ruby On Rails, PostgreSQL e da ferramenta Scrum foram importantes e decisivos na escolhas das mesmas, enquanto o aprendizado de Ionic 2 e Angular 2 foi um desafio a mais no projeto.

Houveram dificuldades no desenvolvimento de partes do sistema, como na autenticação com token via API, porém foram todas superadas. O processo de desenvolvimento deste sistema trouxe grande conhecimento e experiência para o autor, contribuindo assim com seu desenvolvimento profissional, com destaque para a criação de aplicações híbridas na plataforma Ionic 2 e no uso da linguagem TypeScript.

Espera-se que este trabalho estimule o surgimento de outras soluções para a resolução dos problemas de estacionamento, o que resultará em maior competitividade no mercado e melhores produtos para os consumidores.

7.1 FUTURAS ATUALIZAÇÕES

O sistema desenvolvido neste projeto possui diversas funcionalidades, porém é expansível e escalável, podendo receber futuras atualizações para adaptar-se ao uso em estacionamentos

reais. Durante o desenvolvimento e testes do sistema, algumas sugestões foram recebidas e foram constatadas outras possibilidades de atualização, listadas a seguir:

- Permitir a cobrança proporcional aos minutos estacionados, como alternativa aos tickets com tempo pré-determinado;
- Geração de relatórios através da área administrativa;
- Adicionar funcionalidade que permita ao cliente marcar o local em que estacionou, utilizando o GPS do dispositivo móvel;
- Adicionar suporte a diferentes provedores de serviço de pagamento online.

REFERÊNCIAS

- CORDOVA, A. **Apache Cordova**. 2015. Disponível em: <<https://cordova.apache.org/>>. Acesso em: 17/11/2015.
- DESENVOLVIMENTOÁGIL.COM.BR. **Scrum: metodologia ágil para gestão e planejamento de projetos**. 2011. Disponível em: <<http://www.desenvolvimentoagil.com.br/scrum/>>. Acesso em: 15/11/2015.
- FLUIDUI. **Fluid UI**. 2015. Disponível em: <<https://www.fluidui.com/>>. Acesso em: 22/11/2015.
- GOOGLE. **Material Design - Introduction**. 2015. Disponível em: <<https://www.google.com/design/spec/material-design/introduction.html>>. Acesso em: 12/11/2015.
- GROUP, T. P. G. D. **PostgreSQL: Sobre**. 2016. Disponível em: <<https://www.postgresql.org/about/>>. Acesso em: 11/11/2016.
- IONIC. **Ionic 2**. 2016. Disponível em: <<http://ionic.io/2>>. Acesso em: 07/11/2016.
- JR, C. C.; BARAZI, R. A. **Rails 3 Básico**. São Paulo: Novatec Editora, 2011.
- MATERIALIZE. **Materialize**. 2016. Disponível em: <<http://materializecss.com/>>. Acesso em: 07/11/2016.
- MOBILICIDADE. **Mobilicidade - Estacionamento Eletrônico**. 2015. Disponível em: <<http://www.mobilicidade.com.br/siteoficial/estacionamentoeletronico.aspx>>. Acesso em: 15/09/2015.
- MYSQL. **MySQL Workbench**. 2015. Disponível em: <<https://www.mysql.com/products/workbench/>>. Acesso em: 22/11/2015.
- PANGO. **Sistema Pango no Brasil**. 2014. Disponível em: <<http://mypango.com.br>>. Acesso em: 08/09/2015.
- PEREIRA, M. H. R. **AngularJS - Uma abordagem prática e objetiva**. São Paulo: Novatec Editora, 2015.
- RUBY. **Linguagem de programação Ruby**. 2015. Disponível em: <<https://www.ruby-lang.org/pt/>>. Acesso em: 12/11/2015.
- SASS. **Sass: Folhas de Estilo Sintaticamente Impressionantes**. 2016. Disponível em: <<http://sass-lang.com/>>. Acesso em: 01/11/2016.
- SCHWABER, K.; SUTHERLAND, J. **Scrum: metodologia ágil para gestão e planejamento de projetos**. [S.l.]: Scrum.org, 2011.
- SILVA, M. S. **Javascript - Guia do Programador**. São Paulo: Novatec Editora, 2010.

SILVA, M. S. **HTML 5**. São Paulo: Novatec Editora, 2011.

SILVA, M. S. **CSS 3**. São Paulo: Novatec Editora, 2012.

STOPFACIL. **Stopfacil Gerenciador de Estacionamentos**. 2014. Disponível em: <<http://www.mobilicidade.com.br/siteoficial/estacionamentoeletronico.aspx>>. Acesso em: 08/09/2015.

TYPESCRIPT. **Typescript: Javascript que escala**. 2016. Disponível em: <<https://www.typescriptlang.org/>>. Acesso em: 01/11/2016.

VVALE. **ESTAR: Sistema digital entra em aplicação no dia 1º**. 2014. Disponível em: <<http://www.vvale.com.br/geral/estar-sistema-digital-entra-em-aplicacao-dia-1o/>>. Acesso em: 14/09/2015.

VVALE. **UNIÃO DA VITÓRIA: Estacionamento rotativo deixa de funcionar hoje**. 2015. Disponível em: <<http://www.vvale.com.br/geral/uniao-da-vitoria-estacionamento-rotativo-deixa-de-funcionar-hoje/>>. Acesso em: 08/10/2015.

WILLIAM, D. **A História do HTML**. 2012. Disponível em: <<http://www.frontendbrasil.com.br/artigos/a-historia-do-html/>>. Acesso em: 02/11/2016.

APÊNDICE A – PESQUISA

Entre os dias 20 e 24 de novembro de 2015 foi realizada uma pesquisa com 46 pessoas para avaliar o interesse dos potenciais usuários do aplicativo para pagamento de estacionamentos regulamentados. Um formulário foi distribuído na internet através de redes sociais para moradores de Guarapuava-PR e acadêmicos e servidores da UTFPR - Câmpus Guarapuava.

Entre os assuntos perguntados estão as dificuldades encontradas para pagar pelo estacionamento, problemas com a forma de cobrança atual e opções de funcionalidades do aplicativo para avaliar interesse e eventualmente coletar sugestões. A resposta às questões era opcional, ficando a critério do entrevistado escolher as questões que quisesse responder. Os gráficos a seguir mostram as respostas:

Você utiliza automóvel para se locomover pela cidade? (46 respostas)

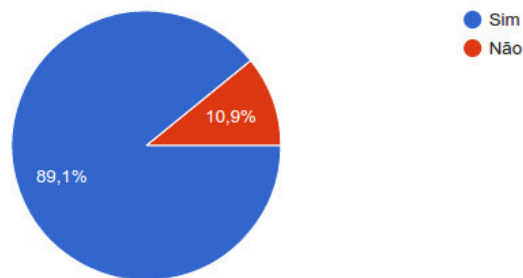


Figura 24: Respostas da pergunta: Você utiliza automóvel para se locomover pela cidade?

Você encontra dificuldades em encontrar lugar para estacionar no centro da cidade?

(45 respostas)

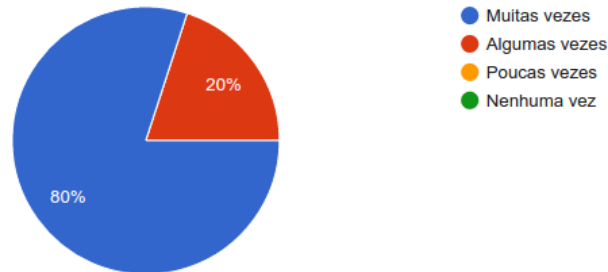


Figura 25: Respostas da pergunta: Você encontra dificuldades em encontrar lugar para estacionar no centro da cidade?

Numa escala de 1 a 5, como você avalia a forma de cobrança atual do estacionamento regulamentado em Guarapuava?

(45 respostas)

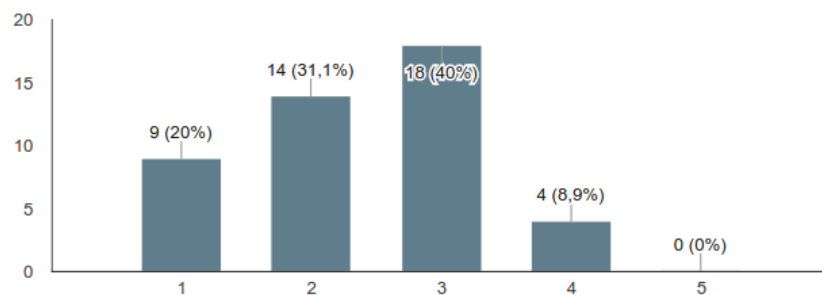


Figura 26: Respostas da pergunta: Numa escala de 1 a 5, como você avalia a forma de cobrança atual do estacionamento regulamentado em Guarapuava?

Você já recebeu pontos na carteira por não saber que foi multado pelo EstaR?

(44 respostas)

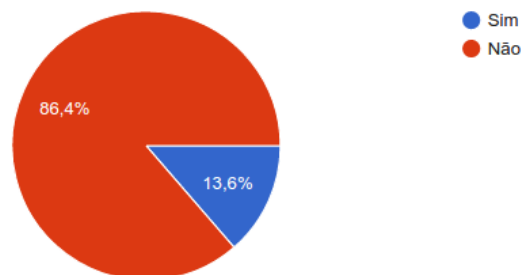


Figura 27: Respostas da pergunta: Você já recebeu pontos na carteira por não saber que foi multado pelo EstaR?

Você utilizaria um aplicativo para Smartphone para pagar pelo uso do estacionamento regulamentado?
(45 respostas)

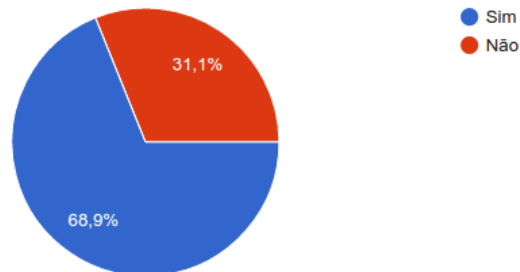


Figura 28: Respostas da pergunta: Você utilizaria um aplicativo para Smartphone para pagar pelo uso do estacionamento regulamentado?

Quais das seguintes formas de cobrança você prefere? (45 respostas)

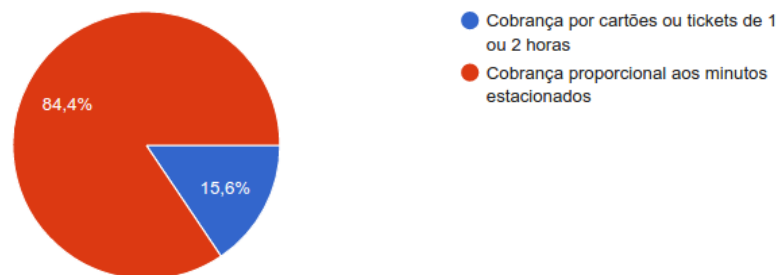


Figura 29: Respostas da pergunta: Quais das seguintes formas de cobrança você prefere?

Qual destas formas de compra de créditos é a melhor para você? (44 respostas)

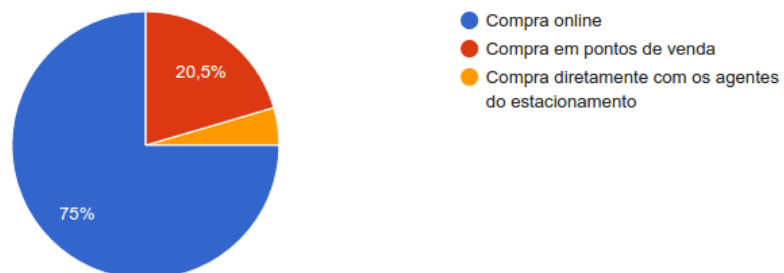


Figura 30: Respostas da pergunta: Quais destas formas de compra de créditos é a melhor para você?