

**UNIVERSIDADE REGIONAL DE BLUMENAU**  
**CENTRO DE CIÊNCIAS EXATAS E NATURAIS**  
**CURSO DE SISTEMAS DE INFORMAÇÃO – BACHARELADO**

**ESTACIONE: PROTÓTIPO DE APLICATIVO PARA  
PAGAMENTO MÓVEL DE ESTACIONAMENTO**

**PAULO ARNOLDO KOGLIN JUNIOR**

**BLUMENAU**  
**2018**

**PAULO ARNOLDO KOGLIN JUNIOR**

**ESTACIONE: PROTÓTIPO DE APLICATIVO PARA  
PAGAMENTO MÓVEL DE ESTACIONAMENTO**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Sistemas de Informação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Profa. Simone Erbs da Costa, Especialista - Orientadora

**BLUMENAU  
2018**

# **ESTACIONE: PROTÓTIPO DE APLICATIVO PARA PAGAMENTO MÓVEL DE ESTACIONAMENTO**

Por

**PAULO ARNOLDO KOGLIN JUNIOR**

Trabalho de Conclusão de Curso aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II pela banca examinadora formada por:

Presidente: \_\_\_\_\_  
Profa. Simone Erbs da Costa, Especialista – Orientadora, FURB

Membro: \_\_\_\_\_  
Prof. Francisco Adell Péricas, Mestre – FURB

Membro: \_\_\_\_\_  
Prof. Mauro Marcelo Mattos, Doutor – FURB

Blumenau, 12 de julho de 2018

Dedico este trabalho especialmente à minha mãe e minha namorada, aos meus amigos mais próximos, minha orientadora e todos que contribuíram de alguma forma para a realização deste trabalho.

## **AGRADECIMENTOS**

À minha mãe, por sempre me incentivar lutar até o fim e apoiar nas horas difíceis.

À minha namorada, por estar sempre do meu lado, me apoiando e ajudando em tudo que estava ao seu alcance.

Aos meus amigos, que me apoiaram e ajudaram durante o desenvolvimento do trabalho.

À minha orientadora, que me ajudou a encarar esse desafio e contribuiu para que esse trabalho obtivesse êxito.

À Deus.

Quanto maiores são as dificuldades a vencer,  
maior será a satisfação.

Cícero

## RESUMO

Este trabalho tem como finalidade apresentar a especificação, desenvolvimento e operacionalidade de um aplicativo móvel (app), intitulado ESTACIONE, que tem como objetivo possibilitar a busca por estacionamentos mais próximos da localização desejada pelo utilizador, bem como efetuar o pagamento por meio de dispositivo móvel. O aplicativo também oferece ao usuário a possibilidade de comparar o preço de outros estacionamentos por meio de um mapa, com o intuito de proporcionar comodidade e agilidade aos seus utilizadores. O app foi desenvolvido sob uma arquitetura cliente-servidor. Na camada cliente, a interface com o usuário foi implementada com base no *framework* Ionic e foi utilizado as linguagens de programação JavaScript, Cascading Style Sheets (CSS) e HyperText Markup Language (HTML) 5. No lado da camada servidor, foi utilizada a *framework* Spring Boot em conjunto com a linguagem de programação Java. A camada servidora foi hospedada na plataforma *Cloud* Heroku, em conjunto com o banco de dados estruturado PostgreSQL. Por fim, foi realizada uma avaliação de usabilidade com base no Método Relationship of M3C with User Requirements and Usability and Communicability Assessment to groupware (M3C-URUCAg) junto a um grupo de pessoas, sendo possível inferir que o estudo realizado atingiu os objetivos propostos, principalmente, devido a satisfação demonstrada pelos resultados obtidos na avaliação, e, por conseguinte, foi possível identificar pontos de melhorias e levantar possíveis extensões.

Palavras-chave: Estacionamento. Pagamento móvel. QR code. Google maps. M3C-URUCAg.

## **ABSTRACT**

This paper aims to present the specification, development and operation of a mobile app called ESTACIONE, which aims to enable the search for parking spaces closer to the location desired by the user, as well as to make the payment through a mobile device. The application also offers the user the possibility of comparing the price of other parking lots by means of a map, to provide comfort and agility to its users. The app was developed under a client-server architecture. In the client layer, the user interface was implemented based on the Ionic framework and we used the programming languages JavaScript, Cascading Style Sheets (CSS) and HyperText Markup Language (HTML) 5. On the server layer side, we used the framework Spring Boot in conjunction with the Java programming language. The server layer was hosted on the Cloud Heroku platform in conjunction with the PostgreSQL structured database. Finally, a usability assessment was carried out with a group of people based on the Relationship between M3C with User Requirements and Usability and Communicability Assessment to groupware (M3C-URUCAg), and it was possible to infer that the study reached the proposed objectives, mainly due to the satisfaction shown by the results obtained in the evaluation, and, therefore, it was possible to identify improvement points and to reach possible extensions.

Key-words: Parking. Mobile payment. QR code. Google maps. M3C-URUCAg.



## LISTA DE FIGURAS

Figura 1 - Quantidade de transações digitais em bilhões por região entre 2011-2015.....	19
Figura 2 - Funcionamento do GPS .....	20
Figura 3 - Evolução dos símbolos .....	20
Figura 4 - Estrutura do símbolo QR Code.....	21
Figura 5 - Pesquisa de estacionamento em um local específico.....	25
Figura 6 - Pesquisa de estacionamento por localização atual.....	26
Figura 7 – Tela para compra de créditos no aplicativo Vaga Inteligente .....	27
Figura 8 - Mapa para pesquisa no aplicativo Vaga Inteligente .....	28
Figura 9 - Mapa do aplicativo ParkMe.....	29
Figura 10 - Diagrama de Casos de Uso do app Estacione.....	35
Figura 11 - Diagrama de Classes do app Estacione.....	36
Figura 12 - Modelo entidade relacionamento do app Estacione.....	38
Figura 13 - Diagrama de Sequência - realizar uma estadia no estacionamento.....	39
Figura 14 - Diagrama de Deploy .....	40
Figura 15 - Plataforma Heroku .....	41
Figura 16 – Ferramenta Eclipse.....	42
Figura 17 – Ferramenta Webstorm .....	43
Figura 18 - Telas de login e inicial do app Estacione .....	49
Figura 19 – Tela do menu do app Estacione.....	50
Figura 20 – Tela de pesquisa no mapa e marcadores dos estacionamentos	51
Figura 21 - Tela para cadastro da forma de pagamento .....	51
Figura 22 - Tela pronta para iniciar uma estadia e mensagem pós início.....	52
Figura 23 - Tela de estadia com estadia em andamento e pós término de estadia .....	53
Figura 24 - Tela de histórico sem nenhuma estadia encontrada e com estadia encontrada.....	54
Figura 25 – Resposta referente ao Gênero dos participantes.....	57
Figura 26 – Resposta referente a Faixa etária dos participantes .....	58
Figura 27 – Resposta referente à Área de atuação dos avaliadores.....	59

Figura 28 - Resultado da pergunta 1.....	61
Figura 29 - Resultado da pergunta 2.....	62
Figura 30 - Resultado da pergunta 3.....	62
Figura 31 - Resultado da pergunta 4.....	63
Figura 32 - Resultado da pergunta 5.....	64
Figura 33 - Resultado da pergunta 6.....	64
Figura 34 - Resultado da pergunta 7.....	65
Figura 35 - Resultado da pergunta 9.....	65
Figura 36 - Resultado da pergunta 8.....	66
Figura 37 - Resultado da pergunta 10.....	66
Figura 38 - Resultado da pergunta 11.....	67
Figura 39 - Resultado da pergunta 12.....	68
Figura 40 - Resultado da pergunta 14.....	68
Figura 41 - Resultado da pergunta 15.....	69
Figura 42 - Resultado da pergunta 16.....	70
Figura 43 - Resultado da pergunta 17.....	70
Figura 44 - Termo de Consentimento Livre e Esclarecido.....	83
Figura 45 - Emoticons aplicados na escala Likert.....	84

## LISTA DE QUADROS

Quadro 1 – Etapas necessárias para realização da avaliação.....	23
Quadro 2 - Escala de severidade.....	23
Quadro 3 - Conjunto básico das Heurísticas de Nielsen .....	24
Quadro 4 - Comparativo entre as características dos trabalhos correlatos .....	30
Quadro 5 - Matriz de rastreabilidade de Requisitos Funcionais X UC .....	36
Quadro 6 – HTML da tela de estadias .....	44
Quadro 7 - Algoritmo para carregar os estacionamentos .....	44
Quadro 8 – Endpoint estadiaAberta do controlador de estadias: backend .....	45
Quadro 9 - Função findAbertaUsuario: back-end.....	45
Quadro 10 - Método do front-end para iniciar uma estadia.....	46
Quadro 11 - Endpoit para iniciar uma estadia no back-end.....	46
Quadro 12 - Função do back-end para iniciar estadia .....	47
Quadro 13 - Função do back-end para finalizar uma estadia .....	47
Quadro 14 - Função do back-end para realizar o pagamento da estadia .....	48
Quadro 15 - Comparativo dos correlatos perante o trabalho desenvolvido .....	55
Quadro 16 – Relação das Heurísticas de Nielsen x Perguntas .....	59
Quadro 17 – Relação das Heurísticas e problemas encontrados.....	60
Quadro 18 - Tabela usuario .....	76
Quadro 19 - Tabela estacionamento .....	76
Quadro 20 - Tabela estadia .....	77
Quadro 21 - Tabela forma_pagamento.....	77
Quadro 22 - Descrição do caso de uso UC01 - Criar usuário .....	78
Quadro 23 - Descrição do caso de uso UC02 - Realizar login.....	78
Quadro 24 - Descrição do caso de uso UC04 - Manter forma de pagamento.....	79
Quadro 25 - Descrição do caso de uso UC05 - Manter perfil .....	79
Quadro 26 - Descrição do caso de uso UC06 - Registrar entrada e saída.....	79
Quadro 27 - Tarefas da Avaliação de Usabilidade e UX pelo Método M3C- URUCAg.....	81

## **LISTA DE ABREVIATURAS E SIGLAS**

ABNT – Associação Brasileira de Normas Técnicas

API – Application Programming Interface

APP – Aplicativo

BD – Banco de Dados

CRUD – Create, Read, Update e Delete

CSS – Cascading Style Sheets

DCU – Diagrama de Casos de Uso

EUA – Estados Unidos da América

GPS – Sistema de Posicionamento Global

HTML – HyperText Markup Language

IDE – Integrated Development Environment

ISO – International Organization for Standardization

JSON – JavaScript Object Notation

M3C-URUCAg – Relationship of M3C with User Requirements and Usability and Communicability Assessment to groupware

MER – Modelo Entidade Relacionamento

QR CODE – Quick Response Code

RF – Requisitos Funcionais

RNF – Requisitos Não Funcionais

SDK – Software Development Kit

UC – Caso de Uso

UML – Unified Modeling Language

UX – Experiência de Usuário

## XP – Extreme Programming

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
1.1 OBJETIVOS.....	16
1.2 ESTRUTURA.....	16
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>18</b>
2.1 PAGAMENTO MÓVEL ( <i>MOBILE PAYMENT</i> ).....	18
2.2 RECURSOS TECNOLÓGICOS.....	19
2.2.1 Sistema de Posicionamento Global.....	19
2.2.2 Quick Response Code .....	20
2.3 USABILIDADE E EXPERIÊNCIA DE USUÁRIO .....	22
2.4 TRABALHOS CORRELATOS.....	24
2.4.1 As tecnologias móveis no auxílio ao estacionamento urbano: o caso de Aveiro.....	25
2.4.2 Vaga Inteligente .....	26
2.4.3 ParkMe .....	28
2.4.4 Correlação entre os trabalhos relacionados.....	29
<b>3 DESENVOLVIMENTO.....</b>	<b>32</b>
3.1 LEVANTAMENTO DE INFORMAÇÕES .....	32
3.1.1 Requisitos.....	32
3.2 ESPECIFICAÇÃO .....	34
3.2.1 Diagrama de casos de uso .....	34
3.2.2 Rastreabilidade dos Requisitos Funcionais e dos Casos de Uso (UC).....	35
3.2.3 Diagrama de Classes .....	36
3.2.4 Modelo entidade relacionamento .....	37
3.2.5 Diagrama de sequência .....	39
3.3 IMPLEMENTAÇÃO .....	39
3.3.1 Técnicas e ferramentas utilizadas.....	39
3.3.2 Implementação do desenvolvimento.....	43
3.3.3 Operacionalidade da Implementação .....	49
3.4 RESULTADOS E DISCUSSÕES.....	54
3.4.1 Comparação entre o app Estacione desenvolvido e os trabalhos correlatos .....	54
3.4.2 Avaliação de usabilidade e experiência de usuário.....	56
<b>4 CONCLUSÕES.....</b>	<b>71</b>

4.1 EXTENSÕES .....	71
<b>REFERÊNCIAS .....</b>	<b>73</b>
<b>APÊNDICE A – DICIONÁRIO DE DADOS .....</b>	<b>76</b>
<b>APÊNDICE B – DESCRIÇÃO DOS CASOS DE USO .....</b>	<b>78</b>
<b>APÊNDICE C – TAREFAS DA AVALIAÇÃO .....</b>	<b>80</b>
<b>APÊNDICE D – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)</b> <b>.....</b>	<b>83</b>
<b>APÊNDICE E – TAREFAS DA AVALIAÇÃO .....</b>	<b>84</b>

## 1 INTRODUÇÃO

Nos últimos anos tem sido observado um aumento significativo na quantidade de automóveis por habitante e um uso excessivo do automóvel (COSTA, 2008). Além disso, o ritmo de vida dos cidadãos que precisam se deslocar diariamente para o trabalho, faculdade, academia, compras entre outros afazeres, está fazendo com que a procura por vagas de estacionamento fique cada vez mais difícil (ALHAK, 2011). Segundo dados do Instituto Brasileiro de Geografia Estatística (IBGE), em 2016 o Brasil totalizava o número de 51.296.981 milhões de automóveis (INSTITUTO BRASILEIRO DE GEOGRAFIA ESTATÍSTICA - IBGE, 2016).

Além do crescente número de automóveis, a população dos mercados em desenvolvimento, como Brasil, Índia e China estão cada vez mais passando para a classe média e adotando soluções móveis, demandando meios inovadores de efetuar transações financeiras, orientados a tecnologia e de fácil integração (MELLON, 2012). A rotina cada vez mais intensa dos cidadãos tem demandado muito tempo e os aplicativos de pagamento móvel tem proporcionado mais comodidade no dia-dia (FOLHA DE SÃO PAULO, 2013). Em 2013, uma pesquisa divulgada pela Folha de São Paulo (2013) já demonstrava uma tendência no uso de pagamento eletrônico para estacionamentos em centros comerciais na cidade de São Paulo, na qual foi constatado que aproximadamente 62% desses centros comerciais já possuíam um sistema para pagamento eletrônico (FOLHA DE SÃO PAULO, 2013).

No ano de 2015 foram realizadas no Brasil 29 bilhões de transações no-cash (sem dinheiro) de um total mundial de 433,1 bilhões. Esse número fez com que o Brasil ocupasse a quarta posição no ranking de países que mais efetuam transações no-cash no mundo, ficando somente abaixo dos Estados Unidos da América (USA), Eurozone e China (CAPGEMINI; PARIBAS, 2017). Ainda, estima-se que no ano de 2020 o mundo dos pagamentos globais será extremamente diferente do que se vê hoje, de modo que será irreconhecível (MELLON, 2012). Para Mellon (2012), os principais fatores dessas mudanças será o impacto tecnológico, a mudança da expectativa dos clientes, a mudança da demografia global, assim como as mudanças na forma de realizar o comércio global. Desta forma, a tendência é que o Mobile Payment (m-Payment) torna-se o substituto para os meios de pagamento tradicionais, tais como: dinheiro, cheque e até mesmo substituir os cartões de crédito ou débito (PEREZ, 2013).

Nesse sentido, o m-Payment é uma das inovações de comunicação digital móvel que se destacou mundialmente e tornou-se uma tendência por possibilitar pagamentos, transferência de valores remotamente por meio de dispositivos móveis dispensando a utilização de dinheiro



em espécie e de qualquer atuação das instituições bancárias (ABRAHÃO, 2015). Pode-se dizer o m-payment proporciona benefícios relacionados à possibilidade de ganho de tempo, praticidade e novas experiências para os consumidores de aplicativos móveis. Além disso, as operadoras de telecomunicações, bancos e provedores de pagamentos tem a oportunidade de atrair novos clientes e reduzir os seus custos operacionais (ABRAHÃO, 2015).

Diante deste cenário, foi desenvolvido um aplicativo móvel para buscar estacionamentos e possibilitar o pagamento por meio de cartões de crédito, proporcionando dessa forma, comodidade e facilidade para os utilizadores localizarem o melhor estacionamento possível. Este trabalho torna-se relevante pelo fato de possibilitar que os motoristas busquem por um estacionamento próximo ao local de destino ou à sua localização atual e que efetuem o pagamento do estacionamento por meio do aplicativo móvel, possivelmente facilitando a rotina do motorista, que não precisará se preocupar em ter sempre dinheiro em espécie.

## 1.1 OBJETIVOS

Este trabalho teve como objetivo geral desenvolver um aplicativo móvel (app) intitulado Estacione para localizar estacionamentos credenciados, integrado a um sistema de pagamento móvel. Para tal, os objetivos específicos foram:

- a) disponibilizar uma interface, contendo um mapa, que possibilite os usuários buscarem por um lugar para estacionar;
- b) disponibilizar uma interface para credenciamento do estacionamento ao aplicativo;
- c) disponibilizar uma interface contendo as informações necessárias para o utilizador realizar pagamento móvel pelo aplicativo;
- d) fornecer informação ao usuário utilizador referente ao tempo de estadia no estacionamento e o preço previsto.

## 1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos. O primeiro capítulo introduz e apresenta a justificativa do trabalho desenvolvido, intitulado Estacione, bem como os objetivos e a definição de sua estrutura. No segundo capítulo é apresentada a fundamentação teórica utilizada para elaboração do projeto, sendo abordados os conceitos de m-payment, recursos tecnológicos, subdivididos em: Sistema de Posicionamento Global (GPS) e Quick Response Code (QR CODE), bem como a seu crescente uso e os seus benefícios; e a contextualização de Usabilidade. Além disso, o capítulo dois apresenta três trabalhos

correlatos. O terceiro capítulo expõe o desenvolvimento do trabalho, trazendo a especificação dos requisitos, o desenvolvimento e a operacionalidade do aplicativo. Ainda neste capítulo é detalhado o processo de avaliação de usabilidade utilizando partes do Método Relationship of M3C with User Requirements and Usability and Communicability Assessment to groupware (M3C-URUCAg), assim como são realizadas as discussões e apresentados os resultados obtidos durante o processo de busca da solução. Por fim, o quarto capítulo aponta as principais conclusões e as sugestões para futuros trabalhos.

## 2 FUNDAMENTAÇÃO TEÓRICA

O objetivo deste capítulo é explorar os principais assuntos para realização do trabalho desenvolvido. Na seção 2.1 é apresentada a definição do Mobile Payment (*m-payment*) e as suas principais características; a seção 2.2 aborda o serviço de posicionamento global e a tecnologia de códigos de barra Quick Response Code (QR Code); ademais, na seção 2.3 é abordada a avaliação de usabilidade e experiência do usuário; e por fim, na seção 2.4 os trabalhos relacionados são apresentados.

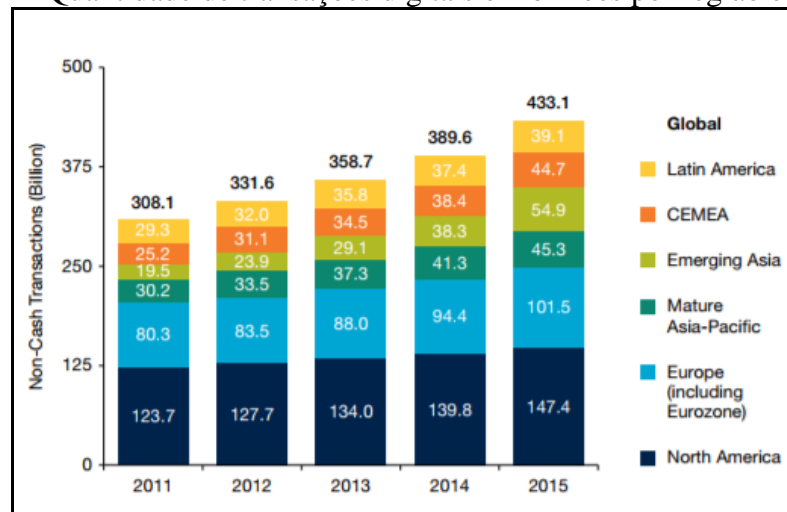
### 2.1 PAGAMENTO MÓVEL (*MOBILE PAYMENT*)

O Mobile Payment (*m-payment*) se caracteriza por ser uma tecnologia que une os dispositivos móveis às transações bancárias, permitindo que as instituições financeiras disponibilizem seus serviços de transações bancárias em tempo real de forma onipresente (HU; LEE; KOU, 2005). Ainda para esses autores, Hu, Lee e Kou (2005), o pagamento móvel possibilita a transferência de valor financeiro entre dois participantes diferentes, sem que exista um contrato entre ambos. Os pagamentos móveis se caracterizam de acordo com o valor da transação bancária. Quando o valor da transação financeira é de aproximadamente 10 dólares consiste em um *micro-payment* (micro pagamento) e acima de 10 dólares, caracteriza-se um *macro-payment* (macro pagamento) (HU; LEE; KOU, 2005).

Segundo Capgemini e Paribas (2017), os volumes globais de transações não monetárias cresceram em torno de 11,2% durante os anos de 2014 e 2015, alcançando o montante de 433,1 bilhões de transações e tornando-se o maior crescimento da última década. Nesse cenário, os cartões de débito e crédito foram os principais meios de pagamentos móveis, enquanto o uso de cheques declinou cada vez mais mundialmente (CAPGEMINI; PARIBAS, 2017).

Nesse sentido, a Figura 1 apresenta a evolução do montante de transações no-cash entre 2011 e 2015, sendo possível observar que houve um aumento contínuo na quantidade de transações (CAPGEMINI; PARIBAS, 2017). O eixo vertical da Figura 1 representa a quantidade em bilhões de transações efetuadas por meio digital por cada região global, enquanto o eixo horizontal da referida figura representa a evolução temporal ao longo dos anos de 2011 até o ano de 2015.

Figura 1 - Quantidade de transações digitais em bilhões por região entre 2011-2015



Fonte: Capgemini e Paribas (2017).

## 2.2 RECURSOS TECNOLÓGICOS

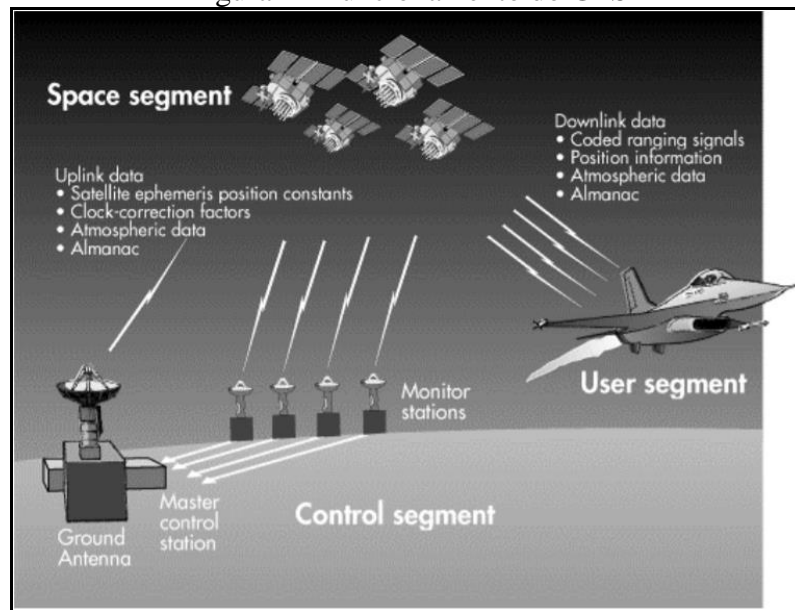
Esta seção traz os dois principais recursos tecnológicos utilizados no desenvolvimento desse trabalho, distribuídos em duas subseções. A subseção 2.2.1 traz o recurso Sistema de Posicionamento Global (GPS) e a subseção 2.2.2 contextualiza Quick Response Code (qr code).

### 2.2.1 Sistema de Posicionamento Global

Segundo Monico (2000), o Sistema de Posicionamento Global (GPS) é um sistema de navegação via rádio desenvolvido pelo Departamento de Defesa dos Estados Unidos da América na década de 70. O GPS funciona por meio do cálculo da distância entre um receptor e outros quatro satélites e consiste a partir de três segmentos, a saber: *space segment*, *control segment* e *user segment*, ilustrados na Figura 2 (BERNARDI; LANDIM, 2002). De acordo com Mai (2014), as partes envolvidas no funcionamento do GPS são:

- a) *space segment* (seguimento espacial): consiste em uma constelação de aproximadamente 24 satélites do governo dos Estados Unidos da América (EUA) distribuídos em seis orbitas. Esses satélites permanecem inclinados a 55° da linha do equador em cerca de 20.200 quilômetros (12.550 milhas) de altitude e circulando totalmente a terra a cada 12 horas;
- b) *control segment* (seguimento de controle): consiste nas estações terrestres que fazem o monitoramento e controle dos satélites GPS;
- c) *user segment* (seguimento de usuário): são os receptores que processam os sinais de navegação dos satélites GPS e calculam o posicionamento e tempo.

Figura 2 - Funcionamento do GPS



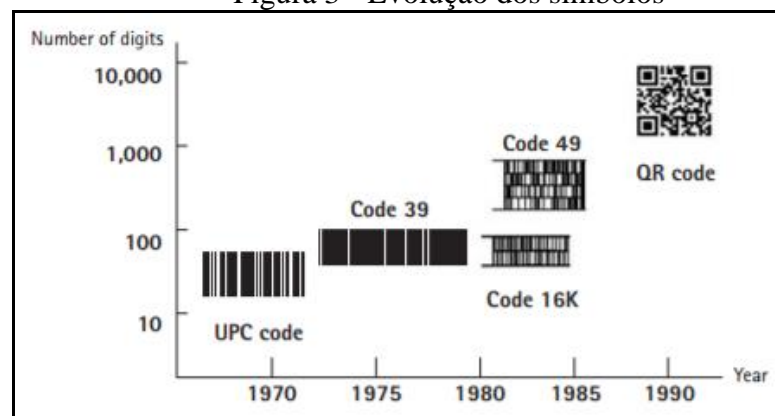
Fonte: Bernardi e Landim (2002).

### 2.2.2 Quick Response Code

No que diz respeito à tecnologia de código de barras em ambientes digitais, cabe destacar o Quick Response Code (QR Code), que é um símbolo bidimensional (2D) criado em 1994 pela empresa Denso, uma das principais empresas do grupo Toyota. Ademais, o QR Code do grupo Toyota possui certificação na norma International Organization for Standardization (ISO) (ISO / IEC18004) em junho do ano do ano de 2000 (SOON, 2008).

Inicialmente, o QR Code foi desenvolvido para ser utilizado no controle de produção de peças automotivas, entretanto, tornou-se aplicável para diversas outras áreas como: varejo; serviços de transporte; pagamentos de tickets de trens e voos entre outros (SOON, 2008). Quando comparado a símbolos lineares, o QR Code pode conter aproximadamente 100 vezes mais dados, conforme demonstra a Figura 3 que contém a evolução dos símbolos.

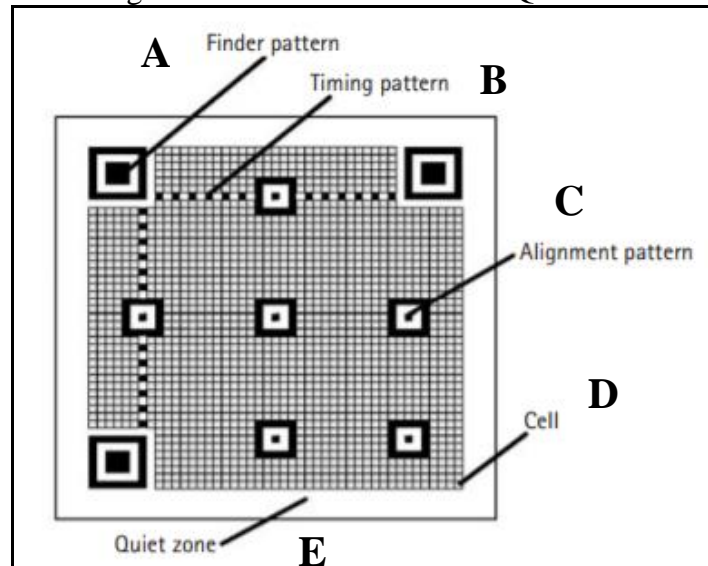
Figura 3 - Evolução dos símbolos



Fonte: Soon (2008).

Na Figura 3, no eixo vertical, enumera-se a quantidade de dígitos que o símbolo suporta. No eixo horizontal da mesma figura, apresenta-se a evolução histórica de cada símbolo. O QR Code, para Soon (2008), consiste em um símbolo de tipo matriz, formado por um conjunto de células dispostas em um quadrado com padrões estruturais para possibilitar um fácil reconhecimento. A Figura 4 demonstra de forma detalhada a estrutura do QR Code.

Figura 4 - Estrutura do símbolo QR Code



Fonte: Soon (2008).

Na Figura 4 ilustra-se as áreas do símbolo destacando cada uma delas com letras, sendo:

- finder pattern* (padrão de busca) (letra A): utilizado para detectar a posição do QR Code, é composto por três quadrados localizados nos cantos das extremidades da matriz. Com essa estrutura o símbolo poder ser detectado em um ângulo de 360°, ou seja, independe da posição que é realizada a leitura do símbolo;
- alignment pattern* (padrão de alinhamento) (letra C): utilizado para corrigir as distorções do QR Code;
- timing pattern* (padrão de temporização) (letra B): utilizado para identificar a localização central de cada célula, representado com um pontilhado preto e branco alternado linearmente em ambas as direções do símbolo.
- quiet zone* (zona silenciosa) (letra E): margem necessária para realizar a leitura do QR Code;
- cell* ou *data area* (célula ou área de dados) (letra D): representado na Figura 4 por toda a área pintada na cor cinza, na qual os dados serão convertidos para binário e armazenados no símbolo.

### 2.3 USABILIDADE E EXPERIÊNCIA DE USUÁRIO

Em geral, quando se fala em usabilidade de softwares ou aplicações, remete-se ao uso de padrões ergonômicos que possam melhorar tanto a consistência da interface, quanto a velocidade que o usuário interage com as aplicações (CYBIS; BETIOL; FAUST, 2015). Neste sentido, pode-se considerar que a usabilidade é a base para que os objetivos sejam atingidos com eficiência e eficácia, e, ainda estabelecendo um vínculo com a Experiência de Usuário (UX), conquanto tenham significados diferentes.

De acordo com a ISO 9241-210, há uma correlação entre a UX e a percepção do utilizador perante o uso da aplicação, bem como a resposta recebida durante o uso (ABNT, 2011). Pode-se observar ainda a equidade da percepção e resposta quanto ao outro objetivo da usabilidade, a satisfação. Ambas abordagens utilizam métodos de avaliação fundamentado na participação do usuário a fim de mensurar o grau de satisfação de um produto. Entretanto, existem dessemelhanças entre essas abordagens, na UX predomina a preocupação com a satisfação, enquanto na usabilidade impera a preocupação perante o desempenho de tarefas (CYBIS; BETIOL; FAUST, 2015).

Em virtude dos avanços tecnológicos, conjuntamente ao uso de dispositivos móveis, pode-se utilizar esses dispositivos a fim de estimular o participante na avaliação de usabilidade. Por conseguinte, quando do uso de um tablet ou um dispositivo móvel, o participante da avaliação tem a possibilidade de responder o questionário em um contexto de interação acerca de sua realidade. Essas técnicas fazem parte de um estudo de Costa (2018), que avalia a usabilidade e comunicabilidade da aplicação pelo método intitulado M3C-URUCAg. Além disso, ao invés de texto, o método utiliza a escala Likert denotada de respostas figuradas por *emoticons*, assim como permite relacionar os requisitos do projeto com a avaliação heurística (COSTA, 2018).

A avaliação heurística, cujas etapas necessárias apresentam-se no Quadro 1, é utilizada em contextos que objetivam identificar problemas relacionados à usabilidade embasados nas melhores práticas definidas por uma série de diretrizes ou heurísticas. De modo a realizar a avaliação, o método pode ser aplicado por profissionais especialistas da área ou da academia, sendo a avaliação realizada por três até cinco especialistas. Ademais, o Quadro 2 traz a escala de severidade encontrada na violação; e o Quadro 3 contempla um conjunto básico das heurísticas de Nielsen.

Quadro 1 – Etapas necessárias para realização da avaliação

ETAPA	SESSÃO	OBSERVAÇÃO	PASSOS
1	Avaliação individual e por especialista em sessões curtas (1 a 2 horas).	Sessões precisam ser individuais para que um avaliador não influencie a opinião dos outros. Em cada sessão de avaliação, o avaliador deve percorrer a interface mais de uma vez para inspecionar os diferentes elementos de interface e compará-los com a lista de heurísticas de usabilidade.	1.1. julga a conformidade da interface com um determinado conjunto de princípios (“heurísticas”) de usabilidade; 1.2. anota os problemas encontrados e sua localização; 1.3. julga a gravidade destes problemas; 1.4. gera um relatório individual com o resultado de sua avaliação e comentários adicionais.
2	Consolidação da avaliação dos especialistas.	Os avaliadores possuem acesso aos relatórios individuais de todos os avaliadores, podendo realizar considerações referente as considerações realizadas pelos demais avaliadores. O artefato resultante dessa etapa é um relatório unificado e consolidado sobre todos os problemas encontrados.	2.1 novo julgamento sobre o conjunto global dos problemas encontrados e 2.2. relatório unificado de problemas de usabilidade
3	Seleção dos problemas a serem corrigidos.	Essa etapa é realizada junto ao cliente (ou gerente de projeto), realizando uma análise de custo/benefício das correções aos problemas encontrados na etapa anterior, levando em consideração a gravidade, os prazos e o orçamento do projeto, bem como a capacitação da equipe de desenvolvimento.	

Fonte: Costa (2018) elaborado de Prates e Diniz (2003).

Quadro 2 - Escala de severidade

Nro	Severidade
0	Não é considerado um problema de usabilidade.
1	Problema apenas visual, não necessita correção imediata.
2	Representa um pequeno problema, a correção deve possuir baixa prioridade.
3	Problema maior de usabilidade, correção deve ser tratada como alta prioridade.
4	Catástrofe de usabilidade, correção deve ser realizada antes da liberação do aplicativo.

Fonte: Costa (2018) elaborado de Moulin (2013).



Quadro 3 - Conjunto básico das Heurísticas de Nielsen

Nro	Heurística	Característica
H1	Visibilidade do estado do sistema	Manter os usuários informados sobre o que está acontecendo, por meio de feedback adequado e no tempo certo.
H2	Correspondência entre o sistema e o mundo real	Utilizar conceitos, vocabulário e processos familiares aos usuários.
H3	Controle e liberdade do usuário	Fornecer alternativas e “saídas de emergência”; possibilidades de <i>undo</i> e <i>redo</i> .
H4	Consistência e padronização	Palavras, situações e ações semelhantes devem significar conceitos ou operações semelhantes; caso haja convenções para o ambiente ou plataforma escolhidos, estas devem ser obedecidas.
H5	Prevenção de erro	Evitar que o erro aconteça, informando o usuário sobre as consequências de suas ações ou, se possível, impedindo ações que levariam a uma situação de erro.
H6	Ajuda aos usuários para reconhecerem, diagnosticarem e se recuperarem de erros	Utilizar mensagens de erro em linguagem simples, sem códigos, indicando precisamente o problema e sugerindo de forma construtiva um caminho remediador.
H7	Reconhecimento em vez de memorização	Tornar os objetos, ações e opções visíveis e compreensíveis.
H8	Flexibilidade e eficiência de uso	Oferecer aceleradores e caminhos alternativos para uma mesma tarefa; permita que os usuários customizem ações frequentes.
H9	Design estético e minimalista	Evitar porções de informação irrelevantes. Cada unidade extra de informação em um diálogo compete com as unidades de informação relevantes e reduz sua visibilidade relativa.
H10	Ajuda e documentação	Facilitar as buscas, focadas no domínio e na tarefa do usuário, e devem listar passos concretos a serem efetuados para atingir seus objetivos.

Fonte: Costa (2018) elaborado de Nielsen (1994).

## 2.4 TRABALHOS CORRELATOS

A seguir são apresentados três trabalhos correlatos relacionados ao projeto proposto. A subseção 2.4.1 aborda o estudo de Ferreira (2014), que consiste no desenvolvimento de um aplicativo móvel intitulado Estacione.ME, para auxiliar no estacionamento urbano de forma ágil e no menor tempo possível. A seção 2.4.2 traz o aplicativo Vaga Inteligente (ESTAPAR, 2018), que possibilita a procura por vagas para estacionar. A seção 2.4.3 apresenta o aplicativo ParkMe Inc. (2016), que facilita a procura de vagas para estacionar. Por fim, a subseção 2.4.4 traz o comparativo entre os trabalhos correlatos.

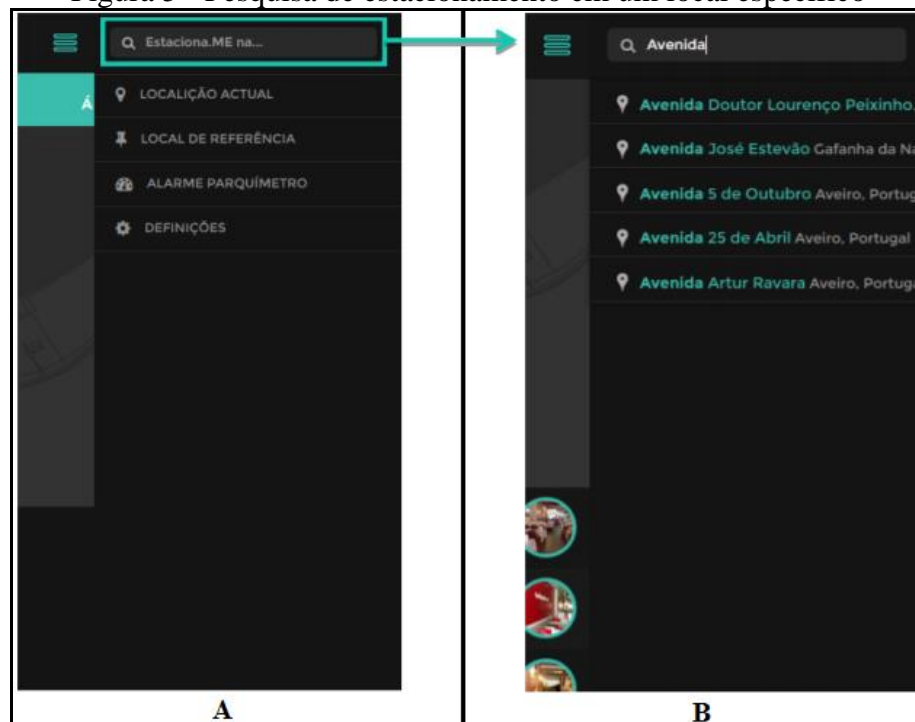
#### 2.4.1 As tecnologias móveis no auxílio ao estacionamento urbano: o caso de Aveiro

Ferreira (2014) teve como objetivo criar um protótipo de um aplicativo móvel para auxiliar no estacionamento urbano da cidade de Aveiro em Portugal, intitulado Estaciona.ME (FERREIRA, 2014). O projeto também teve como objetivo compreender quais metodologias podem auxiliar em um desenvolvimento eficaz em um curto espaço de tempo, assim como os métodos e as ferramentas que possibilitem diminuir o tempo de desenvolvimento de um aplicativo móvel (FERREIRA, 2014).

Ferreira (2014) utilizou a metodologia de desenvolvimento ágil com algumas abordagens das metodologias Extreme Programming (XP) e *Scrum*, por ser ainda a metodologia mais adequada para equipes pequenas desenvolverem em um curto espaço de tempo. A tecnologia utilizada por Ferreira (2014) foi a *framework* PhoneGap (Android), que permite ter acesso a características nativas do dispositivo e por consistir em uma ferramenta *open source* (FERREIRA, 2014).

O aplicativo Estaciona.ME fornece informação para auxiliar um condutor de Aveiro a encontrar um lugar para estacionar seu automóvel, seja ele uma área privada ou em uma rua em que não se pague o parquímetro. Dessa forma, possibilita ao usuário planejar com antecedência em qual local irá estacionar, bem como faz uso de pesquisa: por local específico (Figura 5), por um local de referência e também a partir da localização atual do utilizador (FERREIRA, 2014).

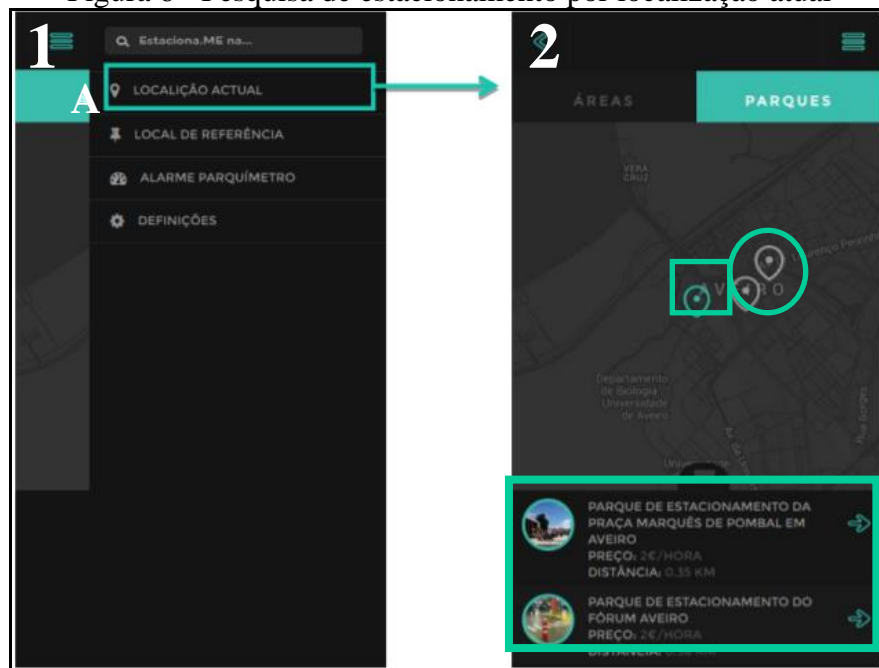
Figura 5 - Pesquisa de estacionamento em um local específico



Fonte: Ferreira (2014).

A Figura 5A se refere a tela para busca de um lugar específico para estacionar. Na parte superior dessa tela pode-se informar de maneira textual o endereço desejado, sendo o resultado da pesquisa apresentado, conforme exemplificado na Figura 5B. Além disso, na Figura 5A também se apresentam outras opções como: a busca pela localização atual, a busca por um local de referência, a opção de alarme do parquímetro e definições do app. Essas opções podem ser observadas na Figura 6, que ilustra uma pesquisa por localização atual (FERREIRA, 2014).

Figura 6 - Pesquisa de estacionamento por localização atual



Fonte: Ferreira (2014).

Na parte superior da tela da Figura 6A, pode-se realizar uma busca pela localização atual. Na tela da Figura 5B, apresenta-se na parte superior os tipos de estacionamentos, destacado no quadrado no centro da tela o marcador com a localização atual do usuário, e ao lado com um círculo, outros dois marcadores com a localização dos estacionamentos. Ademais, na parte inferior apresenta-se em forma de miniatura a foto e resumo de possíveis estacionamentos para selecionar.

#### 2.4.2 Vaga Inteligente

O aplicativo Vaga Inteligente tem como objetivo pagar tíquetes de estacionamentos credenciados ao aplicativo e realizar reservas de vagas em estacionamentos de arenas, aeroportos e etc. (ESTAPAR, 2018). Na Figura 7 é demonstrado que o aplicativo funciona por meio de um sistema de carteira virtual, no qual o usuário precisa adicionar créditos para poder utilizar os serviços disponibilizados pelo app. Ademais, o aplicativo possibilita que o usuário

contrate estacionamentos mensalistas, realizando um único pagamento o direito de usufruir da vaga do estacionamento durante o período estipulado (ESTAPAR, 2018).

Figura 7 – Tela para compra de créditos no aplicativo Vaga Inteligente

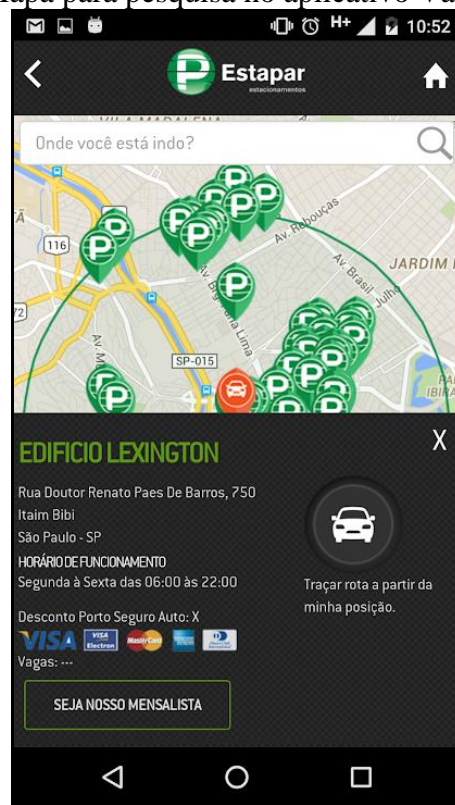


Fonte: Estapar (2018).

Atualmente o aplicativo está presente 75 cidades espalhadas em 13 estados do Brasil, totalizando aproximadamente 400 mil vagas de estacionamento, sendo disponibilizados para as plataformas Android e IOS (ESTAPAR, 2018). O app dispõe de um mapa que permite pesquisar os estacionamentos credenciados (Figura 8), bem como disponibiliza ao usuário informações de todos os estacionamentos que estão próximos do local pesquisado ou também da localização atual do usuário (ESTAPAR, 2018). O aplicativo de Estapar (2018) dispõe das seguintes funcionalidades:

- a) reservar vagas em aeroportos, arenas e estádios;
- b) pagar o estacionamento pelo celular;
- c) recarregar créditos para o estacionamento da Estapar em universidades;
- d) localizar estacionamentos mais próximos do usuário e traçar a rota;
- e) localizar o veículo dentro do estacionamento por meio de QR Code ou GPS;
- f) consultar em tempo real a disponibilidade das vagas na unidade da Estapar desejada;
- g) usufruir de benefícios exclusivos para clientes Porto Seguro Auto.

Figura 8 - Mapa para pesquisa no aplicativo Vaga Inteligente



Fonte: Estapar (2018).

### 2.4.3 ParkMe

O app ParkMe (INRIX INC. 2018) possui como principal objetivo localizar qual o melhor estacionamento para o utilizador, considerando a sua localização atual, assim como disponibilizar informação sobre os preços dos estacionamentos. Além disso, o aplicativo permite que o utilizador reserve e pague antecipadamente por vagas em estacionamentos.

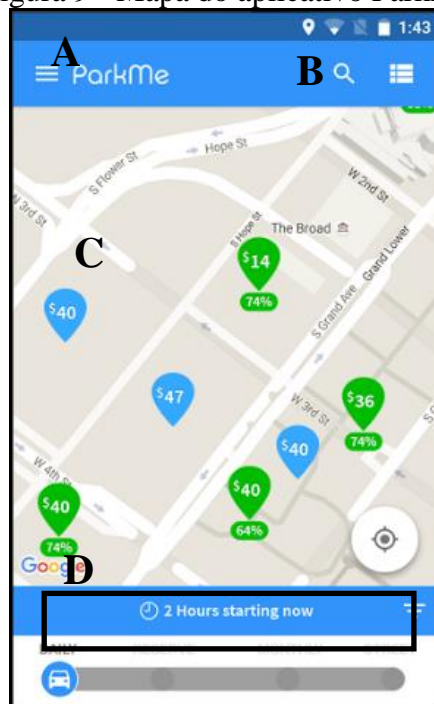
Atualmente o aplicativo móvel está presente em mais de 90 mil endereços em aproximadamente 4.200 cidades em seis continentes e disponível para três as plataformas Android, IOS e Windows Phone. Uma das funcionalidades que o fazem detentor desses números, diz respeito ao aplicativo dispor em um mapa as informações de diversos estacionamentos, destacando o percentual de disponibilidade de vagas abaixo dos marcadores no mapa na Figura 9 (PARKME INC., 2018). Outras funcionalidades também disponibilizadas pelo aplicativo são:

- a) localizar estacionamentos mais baratos e próximos ao destino do utilizador;
- b) disponibilizar informação sobre tarifário dos estacionamentos credenciados;
- c) informar em tempo real a disponibilidade dos estacionamentos;
- d) permitir que o utilizador acompanhe o tempo transcorrido no parquímetro;
- e) oferecer informações sobre locais de interesse perto do estacionamento;

f) traçar a rota até o estacionamento por meio de aplicativos de navegação.

Na tela do app (Figura 9) é disponibilizado o mapa contendo: os marcadores de localização de cada estacionamento, o preço e o percentual de ocupação. O acesso direto ao menu do aplicativo móvel pode ser visualizado na Figura 9 destacado pela letra A. Ao clicar na lupa (letra B da Figura 9), pode-se realizar uma pesquisa por uma localização. Por fim, na letra C, localizada na parte inferior da Figura 9, é possível escolher qual o tipo de busca que o utilizador deseja realizar.

Figura 9 - Mapa do aplicativo ParkMe



Fonte: Inrix Inc. (2018).

#### 2.4.4 Correlação entre os trabalhos relacionados

No Quadro 4 é apresentado um comparativo entre os trabalhos correlatos, as linhas representando as características e as colunas os trabalhos relacionados. Com base no comparativo apresentado no Quadro 4, os trabalhos correlatos trazem o protótipo de Ferreira (2014) e os aplicativos Vaga Inteligente (ESTAPAR, 2018) e ParkMe (INRIX INC, 2018). Os três se destacam por disponibilizarem seus aplicativos para dispositivos móveis e plataforma Android. O aplicativo ParkMe (INRIX INC, 2018) também se destaca por disponibilizar seu aplicativo ainda para as plataformas iOS e Windows Phone, possibilitando atender todo tipo de utilizadores. Já o Vaga Inteligente (ESTAPAR, 2018) ainda disponibiliza seu aplicativo para plataforma IOS.

Quadro 4 - Comparativo entre as características dos trabalhos correlatos

Características \ Correlatos	Ferreira (2014)	Vaga Inteligente Estapar (2018)	ParkMe Inrix Inc. (2018)
Plataforma	Android	Android e IOS	Android, IOS e Windows Phone
Permite pesquisar por estacionamentos.	✓	✓	✓
Disponibiliza informação sobre os estacionamentos mais próximos e mais baratos.	✓	✓	✓
Permite visualizar informações sobre estacionamento livre na rua.	✓	X	✓
Permite visualizar informações sobre estacionamentos privados.	X	✓	✓
Apresenta disponibilidade do lugar de estacionamento em tempo real.	X	✓	✓
Apresenta estacionamentos de qualquer cidade.	X	✓	✓
Permite reservar vaga no estacionamento.	X	✓	✓
Utilizar pagamento via aplicativo móvel.	X	✓	✓

Fonte: elaborado pelo autor.

É importante ressaltar, que a busca por estacionamentos, a localização e o valor a ser pago são informações que trazem comodidade aos motoristas. Nesse sentido, o protótipo de Ferreira (2014) e o aplicativo ParkMe (INRIX INC, 2018) se destacam ao disporem em suas propostas a possibilidade de visualizar informações de estacionamentos livres na rua. Dessa forma, o usuário pode optar por estacionar em um local público ou estacionamento privado.

Os aplicativos ParkMe (INRIX INC, 2018) e Vaga Inteligente (ESTAPAR, 2018) se sobressaem quando comparados ao protótipo de Ferreira (2014), por disponibilizar informações referente aos estacionamentos privados, oferecer ao usuário informações em tempo real da disponibilidade dos estacionamentos de qualquer cidade, assim como possibilitar a reserva de vaga. Outra característica destacada no ParkMe (INRIX INC, 2018) e no Vaga Inteligente (ESTAPAR, 2018) é trazer maior comodidade e segurança para os seus utilizadores, permitindo que o pagamento seja realizado pelo aplicativo móvel. Por outro lado, no trabalho de Ferreira (2014) nota-se um enfoque em resolver problemas no estacionamento urbano.

Tendo em vista os três trabalhos correlatos apresentados, este trabalho apresenta semelhança com os aplicativos ParkMe (INRIX INC, 2018) e Vaga inteligente (ESTAPAR, 2018), principalmente por permitir o pagamento da estadia em estacionamentos privados por meio do aplicativo móvel. Assemelha-se também por auxiliar o usuário a encontrar o estacionamento financeiramente mais acessível e próximo do local desejado, característica essa, que se apresenta no protótipo de Ferreira (2014). Além disso, nota-se relevância devido

ao fato de utilizar técnicas de usabilidade focadas no uso de dispositivos móveis de maneira a proporcionar uma utilização mais confortável ao usuário.

Dessa forma, conjectura-se proporcionar maior comodidade aos utilizadores por meio do pagamento móvel e da localização do melhor estacionamento possível para este, assim como apresentar as contribuições científicas do estudo e as tecnológicas, devido aos recursos computacionais que serão utilizados. Algumas dessas contribuições são:

- a) utilização da tecnologia Quick Response Code (QR CODE) para realizar as marcações de entrada e saída do estacionamento;
- b) utilização dos recursos da *cloud* da Heroku;
- c) utilização de geolocalização para demonstrar o estacionamento mais próximo;
- d) utilização de Application Programming Interface (API) para efetuar transações financeiras.



### 3 DESENVOLVIMENTO

Neste capítulo, são apresentadas as etapas de desenvolvimento do aplicativo desenvolvido, assim como suas especificações e os seus detalhamentos. A seção 3.1 apresenta o levantamento de informações para o desenvolvimento, tais como, os Requisitos Funcionais (RF) e os Requisitos Não Funcionais (RNF). Na seção 3.2 são apresentadas as especificações e a modelagem do app Estacione desenvolvido; enquanto a seção 3.3 detalha o seu desenvolvimento; e por fim, a seção 3.4 apresenta o comparativo entre os trabalhos relacionados e o app Estacione desenvolvido, bem como traz a avaliação de usabilidade e experiência de usuário aplicada aos seus utilizadores; e os resultados obtidos são apresentados e discutidos.

#### 3.1 LEVANTAMENTO DE INFORMAÇÕES

Este trabalho apresenta o desenvolvimento de um aplicativo móvel (app) intitulado Estacione, para auxiliar motoristas de veículos automotores a encontrarem estacionamentos, possibilitando-lhes encontrar o mais próximo, por um preço mais conveniente. O app Estacione possibilita efetuar o pagamento das estadias decorridas em estacionamentos credenciados ao app. Por intermédio de um mapa, o app possibilita ao utilizador encontrar estacionamentos nas proximidades de sua localização atual, bem como próximo de um local pesquisado, dispondo de informações sobre a localização e o preço praticado. Para acessar o app, o utilizador necessita criar uma conta no app e para efetuar pagamentos de estadias via cartão de crédito, necessita prévio cadastramento do cartão de crédito junto à conta criada no app Estacione.

Uma estadia é iniciada a partir do momento que o utilizador efetua a leitura de um QR Code (seção 2.3), que, por conseguinte, dispara um temporizador informando o tempo decorrido e o preço previsto. Para finalizar a estadia, o usuário novamente efetua a leitura do QRCode e o pagamento é enviado à API da Cielo (CIELO, 2018). Além disso, utilizadores podem consultar suas estadias a qualquer momento por meio de um histórico. Desta forma, promove maior controle sobre os gastos e traz facilidade para o dia-a-dia dos utilizadores do app Estacione.

##### 3.1.1 Requisitos

A listagem seguinte contém os requisitos que foram levantados na elaboração deste trabalho. O app Estacione deverá:

- a) exibir um mapa com os estacionamentos credenciados com base na posição atual do usuário (Requisito Funcional - RF);
- b) disponibilizar um cadastro de usuário utilizador do estacionamento (Create, Read, Update e Delete - CRUD) (RF);
- c) disponibilizar um cadastro de estacionamento (CRUD) (RF);
- d) manter um cadastro de perfil do usuário utilizador do estacionamento (CRUD) (RF);
- e) permitir o pagamento do estacionamento pelo aplicativo móvel por meio de um cartão de crédito (RF);
- f) permitir que o usuário vincule um cartão de crédito ao seu cadastro para pagamento do estacionamento (RF);
- g) permitir que o usuário utilizador pesquise por estacionamentos privados credenciados ao aplicativo por um local de referência (RF);
- h) permitir que o usuário utilizador encontre estacionamentos a partir de sua localização atual (RF);
- i) disponibilizar informação sobre o estacionamento, incluindo tarifário praticado pelo estacionamento (RF);
- j) disponibilizar um temporizador a partir do momento que o usuário utilizador der entrada no estacionamento (RF);
- k) realizar a marcação de entrada e saída por meio da leitura de um QR CODE no aplicativo do estacionamento (RNF);
- l) ser implementado na linguagem de programação Javascript, HTML5 para a camada de aplicação (Requisito Não Funcional - RNF);
- m) utilizar Java na implementação do servidor de aplicação gerenciado com Maven (RNF);
- n) utilizar o *framework* Ionic para construção do app Estacione (RNF);
- o) estar disponível na plataforma Android (RNF).
- p) seguir as normas de usabilidade voltadas à aplicativos móveis (RNF);
- q) utilizar o Banco de Dados (BD) PostgreSQL para armazenagem dos dados (RNF);
- r) estar disponível somente online (RNF);
- s) sincronizar com o servidor após cada marcação de entrada e saída (RNF);
- t) utilizar a *cloud* Heroku para o backend do aplicativo (RNF);
- u) utilizar Node.js para gerenciamento da aplicação cliente (RNF).

## 3.2 ESPECIFICAÇÃO

A presente seção aborda a especificação do aplicativo desenvolvido, bem como seu detalhamento e diagramações realizados por meio da ferramenta Astah Community. Na subseção 3.2.1 é apresentado o Diagrama de Casos de Uso (DCU), dando ênfase ao principal Caso de Uso (UC). A descrição dos demais casos de uso encontram-se no Apêndice B, bem como apresenta a matriz de rastreabilidade dos requisitos funcionais e dos casos de uso. Na subseção 3.2.3 é trazida a especificação da aplicação por meio do diagrama de classes. Já a subseção 3.2.4 traz o Modelo Entidade Relacionamento (MER). Por fim, a subseção 3.2.5 apresenta o Diagrama de Sequência do app Estacione.

### 3.2.1 Diagrama de casos de uso

Esta subseção apresenta o Diagrama de Caso de Uso (DCU), demonstrado por meio da Figura 10, que representa as principais funções imputadas aos atores. O ator `USUARIO` corresponde algum usuário utilizador cadastrado no app Estacione, cujo perfil dispõe das funcionalidades para buscar estacionamentos e realizar estadias. Já o ator `ESTACIONAMENTO` representa algum estacionamento que se credenciou à plataforma.

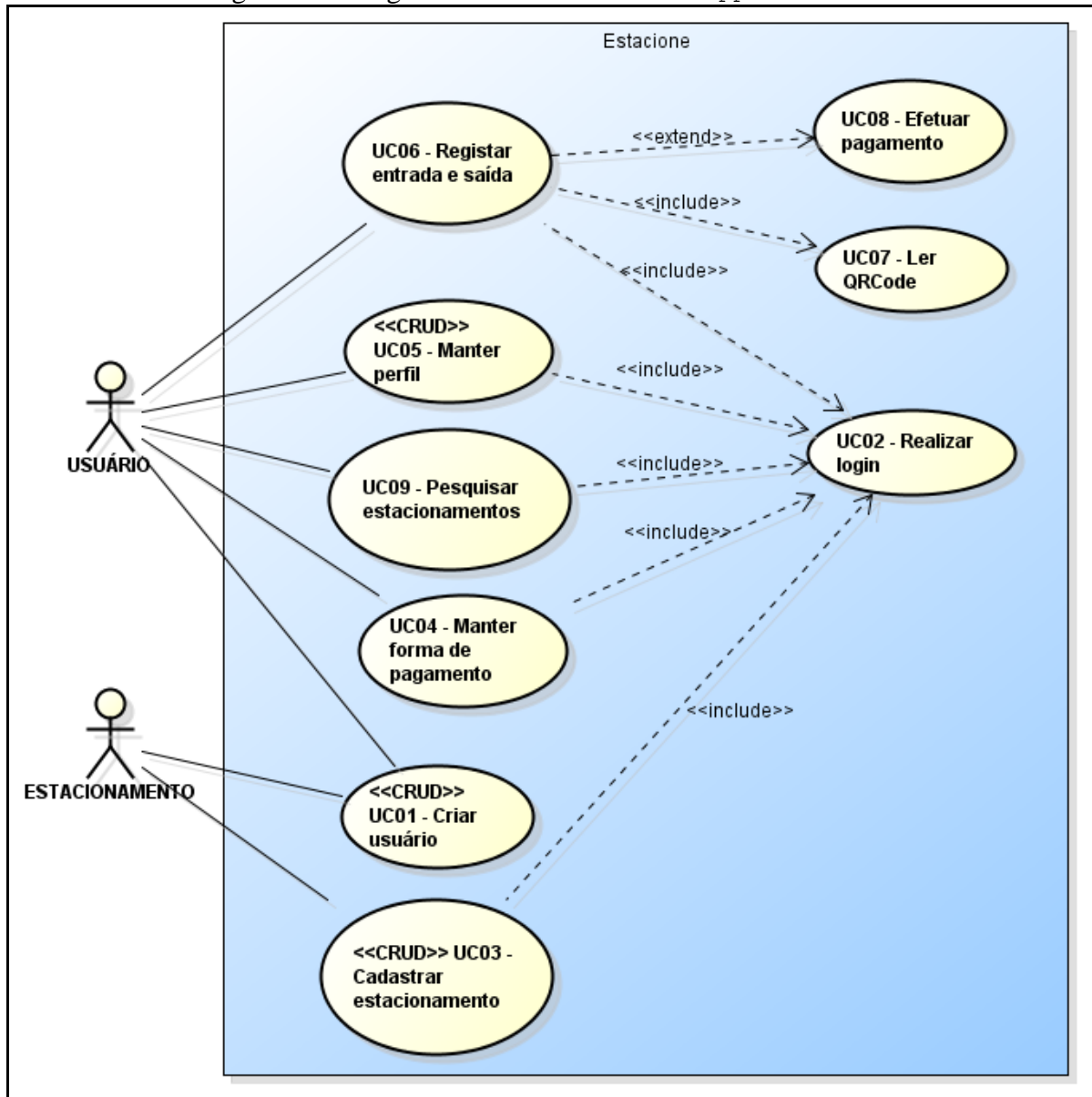
Conforme a Figura 10, ao ator `USUARIO` compete grande parte dos casos de uso. Os atores `USUARIO` e `ESTACIONAMENTO` dispõem do caso de uso `UC01 - Criar usuário`, que consiste no cadastramento do usuário no app Estacione, permitindo posteriormente ingressar no sistema. Uma vez cadastrado, incumbe ao ator `USUARIO` também a realização de realizar o login no app, representado no `UC02 - Realizar login`. Ao abrir o app, sempre é necessário realizar o login, pois conforme representado no diagrama, todos os Casos de Uso (UC) incluem a realização do `UC02 - Realizar login`.

O caso de uso `UC06 - Registrar entrada e saída` é responsável pela principal funcionalidade do app, iniciar uma estadia em um estacionamento, e ainda, finalizar a estadia previamente iniciada. Além disso, esse UC é responsável por contabilizar o valor a ser pago e efetivar o pagamento da estadia junto à plataforma de transações móveis Cielo, representado no caso de uso `UC08 - Efetuar pagamento`. Não obstante, antes de realizar a entrada e a saída, obrigatoriamente se deve fazer a leitura do código de barras QRCode (seção 2.2), procedimento representado no caso de uso `UC07 - Ler QRCode`.

O caso de uso `UC05 - Manter perfil` é incumbido da manutenção dos dados cadastrais do `USUARIO`, do mesmo modo que o caso de uso `UC04 - Manter forma de pagamento` é incumbido da manutenção dos dados do cartão de crédito do usuário.

Já o caso de uso UC03 – Cadastrar estacionamento compete, com exclusividade, ao ator ESTACIONAMENTO, e, consiste no cadastramento das informações do estacionamento, que posteriormente se utiliza para dispor ao ator USUÁRIO no mapa do app. Por fim, o caso de uso UC09 – Pesquisar Estacionamentos, permite o USUARIO buscar por estacionamentos no mapa do aplicativo e exibir as informações do estacionamento ao USUARIO.

Figura 10 - Diagrama de Casos de Uso do app Estacione



Fonte: elaborado pelo autor.

### 3.2.2 Rastreabilidade dos Requisitos Funcionais e dos Casos de Uso (UC)

O Quadro 5 apresenta a matriz de rastreabilidade entre os requisitos especificados na seção 3.1.1 e os casos de uso apresentados na Figura 10.

Quadro 5 - Matriz de rastreabilidade de Requisitos Funcionais X UC

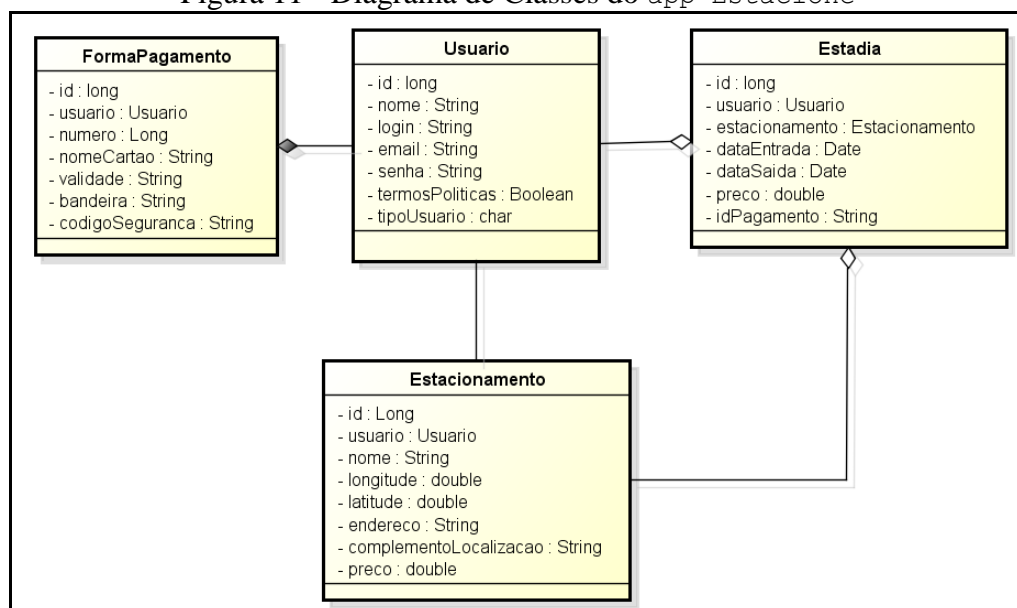
REQUISITOS FUNCIONAIS	UC
RF01: Exibir um mapa com os estacionamentos credenciados com base na posição atual do usuário.	UC09
RF02: Disponibilizar um cadastro de usuário utilizador do estacionamento (CRUD).	UC01
RF03: Disponibilizar um cadastro de estacionamento (CRUD).	UC03
RF04: Manter um cadastro de perfil do usuário utilizador do estacionamento (CRUD).	UC05
RF05: Permitir o pagamento do estacionamento pelo aplicativo móvel por meio de um cartão de crédito.	UC08
RF06: Permitir que o usuário vincule um cartão de crédito ao seu cadastro para pagamento do estacionamento.	UC04
RF07: Permitir que o usuário utilizador pesquise por estacionamentos privados credenciados ao aplicativo por um local de referência.	UC09
RF08: Permitir que o usuário utilizador encontre estacionamentos a partir de sua localização atual.	UC09
RF09: disponibilizar informação sobre o estacionamento, incluindo tarifário praticado pelo estacionamento.	UC09
RF10: disponibilizar um temporizador a partir do momento que o usuário utilizador der entrada no estacionamento.	UC06 e UC07

Fonte: elaborado pelo autor.

### 3.2.3 Diagrama de Classes

A seguir, na Figura 11, são apresentadas as classes equivalentes às principais entidades do aplicativo desenvolvido, assim como os seus respectivos atributos. Desta forma, é possível obter uma visão geral da estrutura do aplicativo e como as principais classes se relacionam.

Figura 11 - Diagrama de Classes do app Estacione



Fonte: elaborado pelo autor.

Na Figura 11, a classe `Usuario` representa um usuário cadastrado no sistema, contendo informações para realizar o *login* no aplicativo. O tipo do usuário é diferenciado por meio do atributo `tipoUsuario`, caracterizando se é um `motorista` ou um `estacionamento`. Essa classe também serve de subsídio para o controle de acesso ao app. A partir dela, o *framework* de autenticação obtém os atributos `login` e `senha` a fim de verificar a validade das credenciais do usuário.

A classe `FormaPagamento` guarda a forma de pagamento cadastrada pelo usuário, informação a qual se utiliza ao finalizar estadias, enviando-as junto à transação de pagamento móvel. A classe possui o atributo `id` para controle interno, o `Usuario` ao qual pertence a forma de pagamento, e os demais atributos para armazenamento do cartão de crédito.

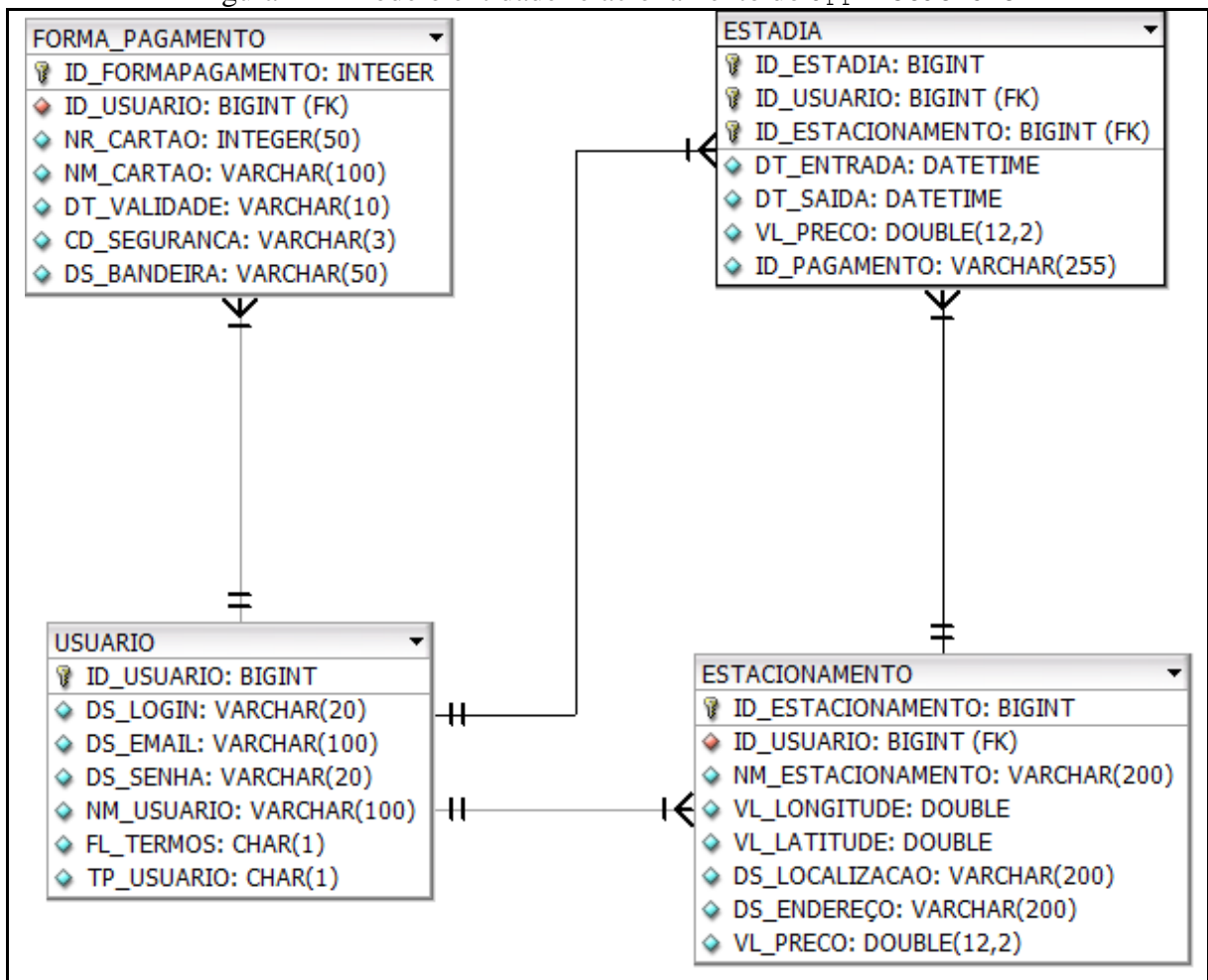
A classe `Estacionamento` representa um estacionamento cadastrado no app, que recebe a identificação do usuário vinculado ao estacionamento por meio do atributo `usuario`. Nesta classe também são armazenados dados como: `longitude` e `latitude`, atributos utilizados para informar a localização do estacionamento no mapa; `nome`, utilizado para guardar o nome do estacionamento; os atributos `endereco` e `complementoLocalizacao`, ambos auxiliares à localização no mapa, e o `preco`, armazenando o valor praticado pelo estacionamento.

Por fim, a classe `Estadia` representa as estadias realizadas por um usuário em um estacionamento qualquer. A classe possui vínculo à classe `Usuario`, bem como à classe `Estacionamento`, respectivamente, pelos atributos `usuario` e `estacionamento`. Na classe `Estadia` também são armazenadas a data da entrada, representada por meio do atributo `dataEntrada`, e a data da saída, representada pelo atributo `dataSaida`, ambas utilizadas para apuração do tempo decorrido no estacionamento, formando assim, a base para cálculo do preço total pago pela estadia, guardado no atributo `preco`. Ademais, contém o atributo `idPagamento` para armazenar a identificação da transação de pagamento, recebida após efetivação do pagamento por meio da API da Cielo.

### 3.2.4 Modelo entidade relacionamento

A Figura 12 apresenta o Modelo de Entidade Relacionamento (MER) utilizado para compor a estrutura de persistência do aplicativo desenvolvido. Para elaboração do MER, foi utilizada a ferramenta DBDesigner Fork. Além de ser grátis, a ferramenta é robusta, suporta as plataformas Windows e Linux, e ainda, possibilita fazer a modelagem de bancos de dados em Oracle, SQL Server, MySQL, Firebird e PostgreSQL (SOURCEFORGE, 2016).

Figura 12 - Modelo entidade relacionamento do app Estacione



Fonte: elaborado pelo autor.

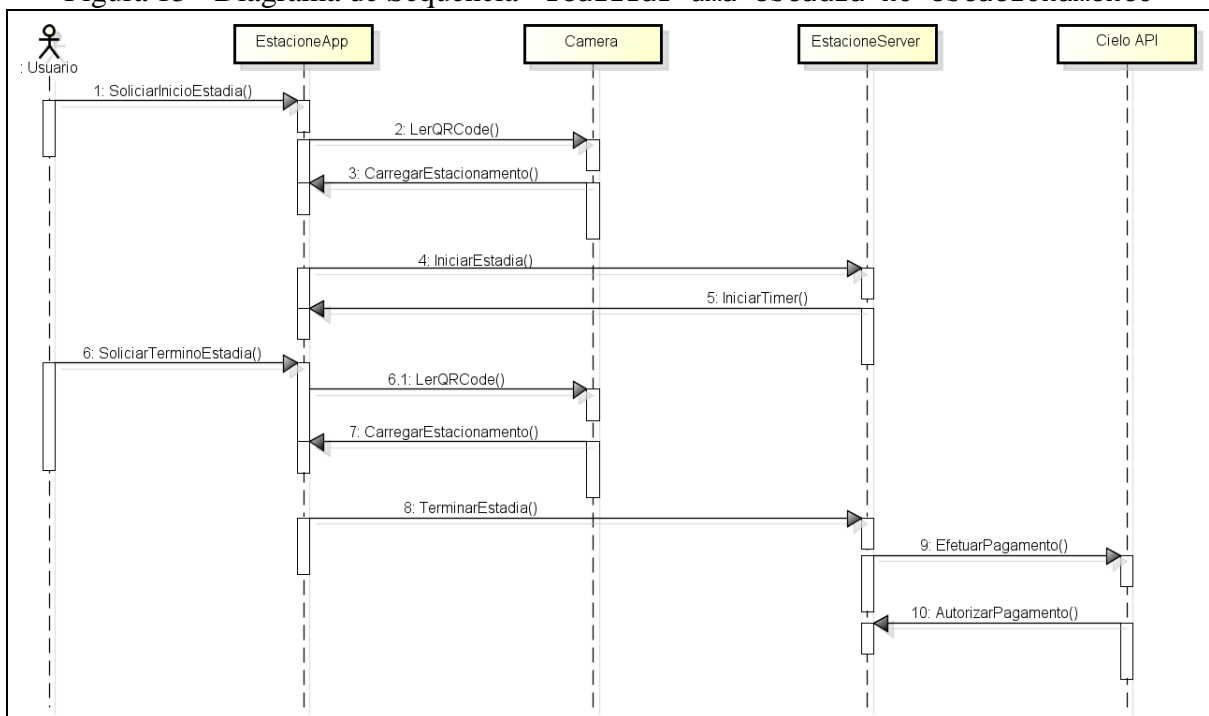
As entidades presentes na Figura 12, utilizadas no desenvolvimento do app Estacione, são definidas brevemente em seguida e o Apêndice A traz detalhadamente a especificação do dicionário de dados utilizado no desenvolvimento do aplicativo:

- a) a entidade `USUARIO` armazena os dados dos usuários quando cadastrados ao criar uma conta;
- b) a entidade `ESTACIONAMENTO` armazena os dados do estacionamento cadastrado pelo usuário, que representa um estacionamento;
- c) a entidade `FORMA_PAGAMENTO` armazena os dados da forma de pagamento do usuário, que posteriormente é utilizada para efetuar o pagamento de suas estadias;
- d) a entidade `ESTADIA` armazena as estadias dos usuários nos estacionamentos, que consiste no ato de marcar a entrada e saída do estacionamento.

### 3.2.5 Diagrama de sequência

A Figura 13 apresenta o diagrama de sequência da principal função do app Estacione, realizar uma estadia no estacionamento. O ator `USUARIO` corresponde ao usuário operando o aplicativo no smartphone, já o `EstacioneApp`, representa o aplicativo instalado no smartphone do `Usuario` e a `Camera`, obviamente, a câmera do celular. Ainda, o `EstacioneServer` representa o servidor backend do aplicativo, ao qual incumbe as regras de negócio e a persistência no banco de dados, e, por fim, a `Cielo API` equivale à administradora de pagamentos móveis, responsável as solicitações de pagamento do app Estacione, junto aos dados do cartão de crédito do `USUARIO`, e, efetuar o pagamento da Estadia concluída.

Figura 13 - Diagrama de Sequência - realizar uma estadia no estacionamento



Fonte: elaborado pelo autor.

## 3.3 IMPLEMENTAÇÃO

Nesta seção são apresentadas as técnicas e ferramentas utilizadas para o desenvolvimento do aplicativo Estacione (subseção 3.3.1), a implementação (subseção 3.3.2), e, por fim, a operacionalidade da implementação (subseção 3.3.3).

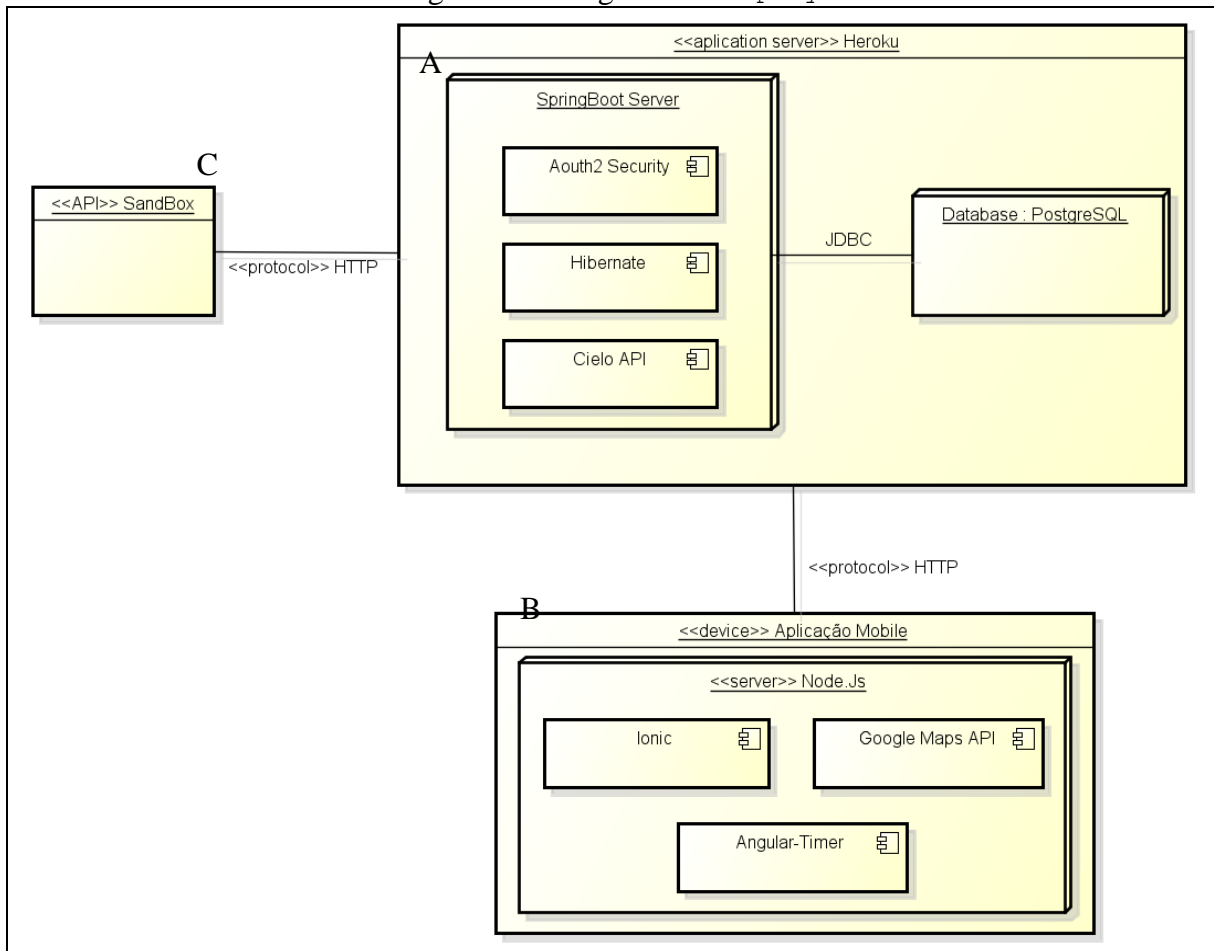
### 3.3.1 Técnicas e ferramentas utilizadas

Foi utilizada uma estrutura cliente-servidor para desenvolver o app Estacione, sendo a camada `cliente` responsável pela interface com o usuário, e a camada `servidor` incumbida



de gerenciar a persistência com o banco de dados e a integração com a plataforma de pagamentos móveis. Além disso, foi utilizada a Unified Modeling Language (UML) para modelagem do app Estacione desenvolvido. A UML utiliza notações gráficas, ajudando na especificação de um projeto de softwares (FOWLER, 2014). A Figura 14 traz o diagrama de deploy do trabalho desenvolvido, demonstrado a arquitetura, os *frameworks*, os principais componentes e as APIs utilizadas.

Figura 14 - Diagrama de Deploy



Fonte: elaborado pelo autor.

Na Figura 14, a letra A representa o servidor de aplicação, construído com Spring Boot<sup>1</sup>, agregando-lhe os frameworks Aouth2<sup>2</sup> para gerenciamento de segurança da camada de comunicação Rest, o Hibernate<sup>3</sup> para comunicação com o banco de dados PostgreSQL<sup>4</sup>, e ainda, a API da Cielo<sup>5</sup> para comunicação com a plataforma de pagamentos móveis. A camada

<sup>1</sup> Disponível em: <<https://spring.io/projects/spring-boot>>. Acesso em: 01 jun. 2018.

<sup>2</sup> Disponível em: <<https://oauth.net/2/>>. Acesso em: 01 jun. 2018.

<sup>3</sup> Disponível em: <<http://hibernate.org/>>. Acesso em: 01 jun. 2018.

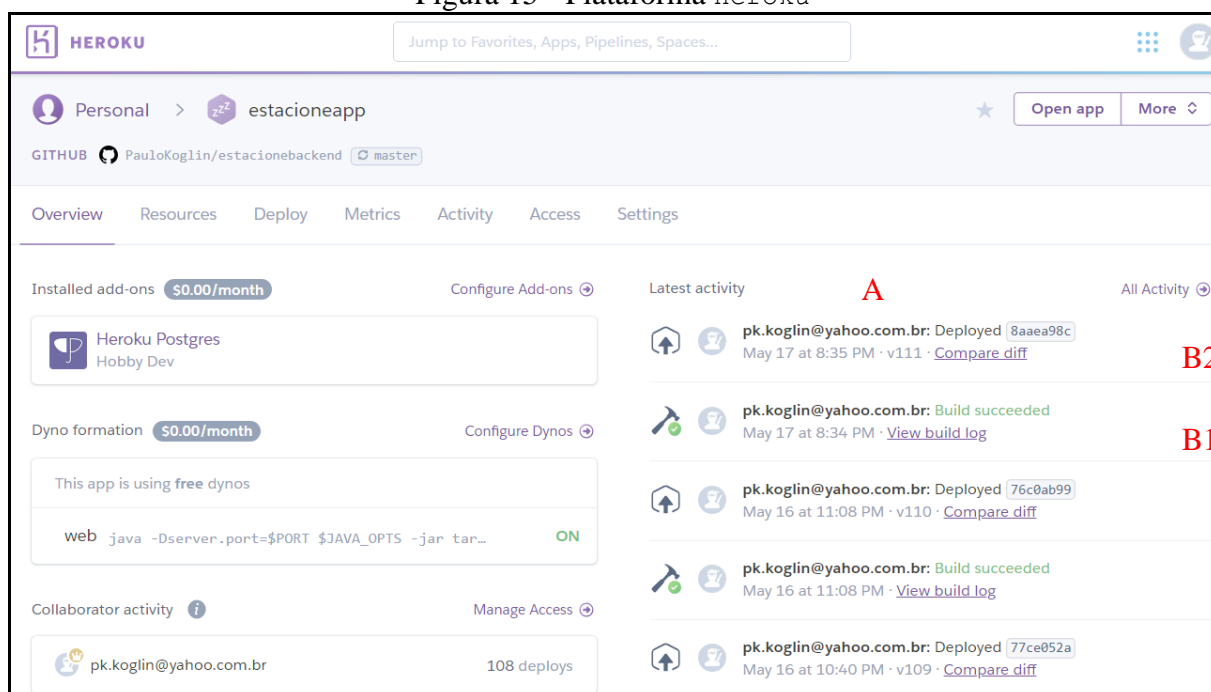
<sup>4</sup> Disponível em: <<https://www.postgresql.org/>>. Acesso em: 01 jun. 2018.

<sup>5</sup> Disponível em: <<https://developercielo.github.io/>>. Acesso em: 01 jun. 2018.

do servidor foi disponibilizada na cloud Heroku<sup>6</sup> (Figura 15), plataforma baseada em um sistema de contêiner gerenciável, que por sua vez, conta com diversos serviços de dados integrados, possibilitando a execução e implementação de aplicativos modernos (SALESFORCE, 2018).

O Heroku possui integração com a plataforma GitHub<sup>7</sup>, possibilitando a automatização do *deploy* sempre que efetuado um push no repositório. Esta funcionalidade foi utilizada para otimizar o desenvolvimento do app Estacione, podendo ser observada na letra A da Figura 15, o histórico das últimas atividades e a letra B1, a compilação do app após o *commit*, e a letra B2 o *deploy*.

Figura 15 - Plataforma Heroku



Fonte: elaborada pelo autor.

Na letra B da Figura 14, é apresentada a estrutura do aplicativo móvel construído sob o *framework* Ionic<sup>8</sup>, juntamente com a Google Maps Javascript API<sup>9</sup>. O *framework* Ionic consiste em um Software Development Kit (SDK) gratuito, permitindo a criação de app de alta qualidade e utilizando as tecnologias HyperText Markup Language (HTML), Cascading Style Sheets (CSS) e JavaScript (IONIC FRAMEWORK, 2018). Já a API Google Maps Javascript API, permite os desenvolvedores customizarem mapas com conteúdo próprio, dispondo tanto de opções para alterar o estilo do mapa, quanto serviços e bibliotecas

<sup>6</sup> Disponível em: <<https://www.heroku.com/>>. Acesso em: 01 jun. 2018.

<sup>7</sup> Disponível em: <<https://github.com/>>. Acesso em: 01 jun. 2018.

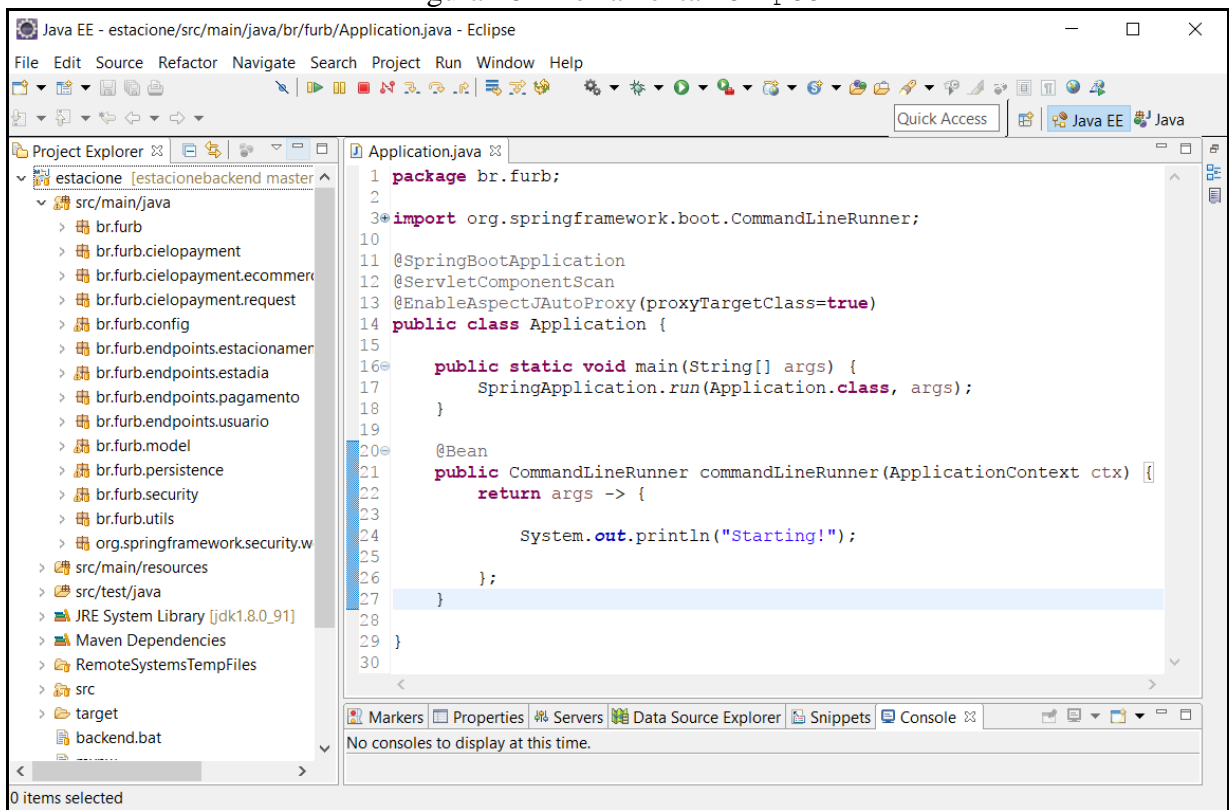
<sup>8</sup> Disponível em: <<https://ionicframework.com/>>. Acesso em: 01 jun. 2018.

<sup>9</sup> Disponível em: <<https://cloud.google.com/maps-platform/>>. Acesso em: 01 jun. 2018.

auxiliares (GOOGLE DEVELOPERS, 2018). Já a letra C apresenta a plataforma Cielo API, que foi utilizada junto à aplicação `servidor` para realizar os pagamentos móveis.

O desenvolvimento do app Estacione foi dividido em duas partes, uma parte para desenvolver a camada `servidor`, representado pelas letras A e C da Figura 14, e outra parte para a camada `cliente`, representado pela letra B da mesma figura. A camada `servidor` (Figura 14, letra A) foi desenvolvida por meio da Ferramenta Eclipse<sup>10</sup> (Figura 16), uma Integrated Development Environment (IDE) da linguagem de programação Java, que possibilita sua personalização por meio de pacotes de extensão (ECLIPSE FOUNDATION, 2018).

Figura 16 – Ferramenta Eclipse



Fonte: elaborado pelo autor.

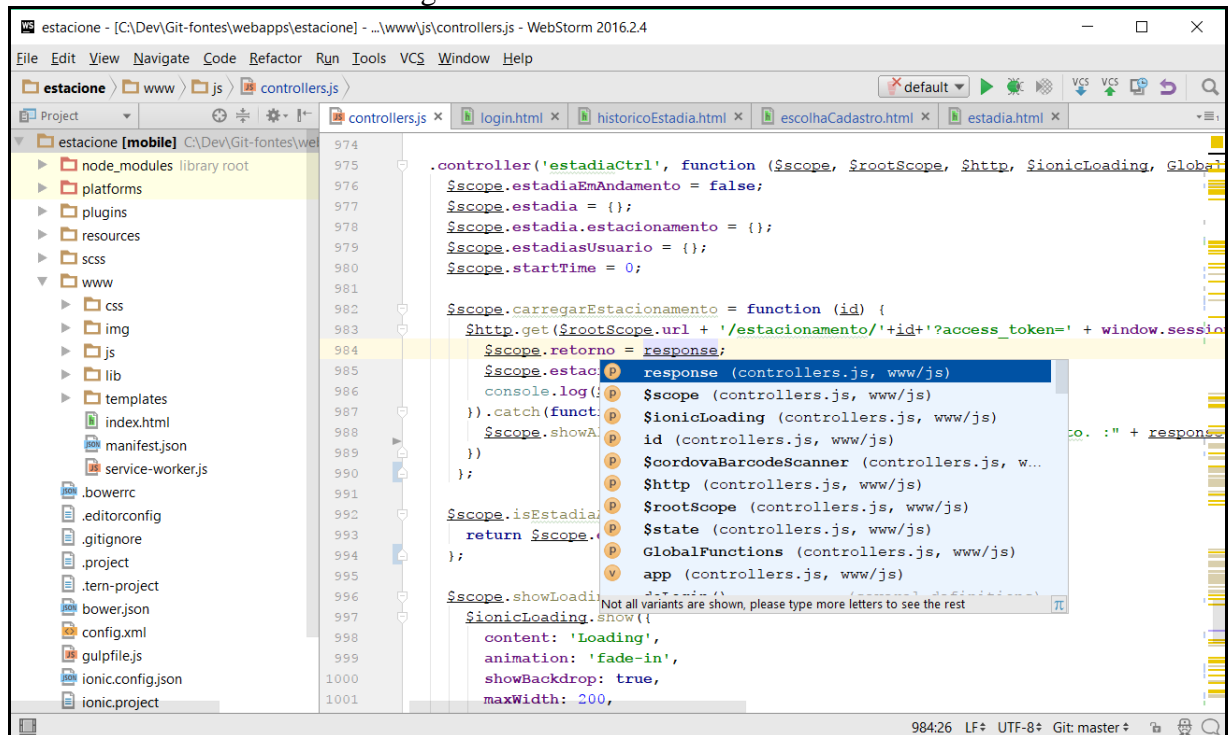
Por outro lado, na camada `cliente` (Figura 14, letra B), foi utilizada a IDE Javascript WebStorm<sup>11</sup> (Figura 17), que conta com algumas funcionalidades, como preenchimento de código inteligente e detecção de erros instantâneo, que auxiliam no desenvolvimento de aplicativos com JavaScript e TypeScript (JET BRAINS, 2018). Diante dessas funcionalidades, a IDE proporciona maior produtividade durante o desenvolvimento e

<sup>10</sup> Disponível em: <<https://www.eclipse.org/>>. Acesso em: 01 jun. 2018.

<sup>11</sup> Disponível em: <<https://www.jetbrains.com/webstorm/>>. Acesso em: 01 jun. 2018.

evita que um erro de sintaxe passe despercebido pelo desenvolvedor. Além disso, a IDE WebStorm também se destaca por disponibilizar aos estudantes de universidades credenciadas à plataforma online, uma conta de estudante, que permite a utilização da IDE sem necessidade de pagamento de licença de uso.

Figura 17 – Ferramenta Webstorm



Fonte: Jet Brains (2018).

### 3.3.2 Implementação do desenvolvimento

Com o objetivo de explicar a fase de desenvolvimento do app Estacione, apresentar-se-á nesta subseção a rotina utilizada para iniciar uma estadia em um estacionamento, tanto na camada cliente (*front-end*), quanto na camada servidor (*back-end*). Preliminarmente, apresenta-se no Quadro 6 o código HTML referente à tela para realização de estadias (subseção 3.2.1, UC06 – Registrar entrada e saída), a qual se encontra na camada cliente (aplicativo móvel). Nesta tela foi utilizado o componente Angular-Timer (subseção 3.3.1, letra B), ao qual incumbe a função de exibir o tempo decorrido na estadia.

No Quadro 6, ao abrir a tela de estadia, para determinar a opção disponível ao usuário, iniciar ou terminar uma estadia, é utilizada a função `isEstadiaAndamento` (linhas 6 e 9), que verifica a variável `estadiaEmAndamento` (linha 1007), que, por sua vez, é inicializada ao abrir a tela por meio da função `verificaEstadiaEmAndamento` (linha 2). Quanto ao tempo decorrido nas estadias, na linha 18 é definida a hora de início do

*timer* por meio da diretiva `start-time`, na qual é atribuída a variável `starttime`, definida posteriormente ao iniciar uma estadia. Ainda, na linha 18, é definida a quantidade de mil milissegundos (um segundo) na diretiva `interval`, indicando assim, o intervalo de tempo que o *timer* incrementará a cada segundo. Além disso, na linha 19 é definido o formato de apresentação do tempo decorrido por meio das variáveis `hhours`, `mminutes` e `sseconds`, representando respectivamente a formatação composta pelas horas, minutos e segundos (`hh:mm:ss`). Já o Quadro 7 traz a função `verificaEstadiaAndamento`.

Quadro 6 – HTML da tela de estadias

```

1 <ion-view view-title="Estadia" id="deviceready" hide-nav-bar="false" class="page" >
2 <ion-content padding="true" scroll="false" ng-controller="estadiaCtrl" ng-init="verificaEstadiaAndamento()" >
3   <div class="row responsive-md">
4     <div class="col col-50 col-offset-25">
5       <div class="button-bar-inline">
6         <button ng-show="isEstadiaAndamento()" class="button button-block button-dark" ng-click="terminarEstadia('poll-server')">
7           Terminar
8         </button>
9         <button ng-show "!isEstadiaAndamento()" class="button button-block button-dark" ng-click="scanNow('poll-server')">
10          Iniciar
11        </button>
12        <div ng-show="isEstadiaAndamento()" class="list-item" timer-controls ng-init="timerType = 'Polling Server'">
13          <h2 class="title" style="color: #eeeeee;">
14            Estadia em andamento
15          </h2>
16          <h4 class="title" style="color: #eeeeee;">
17            Entrada: {{estadia.dataEntrada | date : 'dd/MM/yyyy HH:mm'}}<br>
18            <timer start-time="startTime" interval="1000" auto-start="false">
19              Tempo decorrido : {{hours}}:{{minutes}}:{{seconds}}
20            </timer>
21            <br><span ng-show="estadia.estacionamento.preco > 0">Preço Estimado : {{timerConsole}}</span> </br>
22            <br><span ng-show="estadia.estacionamento.preco == 0">Preço Estimado : Calculando...</span> </br>
23          </h4>
24        </div>
25      </div>
26    </div>
27  </ion-content>
28 </ion-view>

```

Fonte: elaborado pelo autor.

Quadro 7 - Algoritmo para carregar os estacionamentos

```

1006 $scope.isEstadiaAndamento = function () {
1007   return $scope.estadiaEmAndamento;
1008 };
1009
1010 $scope.verificaEstadiaAndamento = function () {
1011   $scope.showLoading();
1012   console.log("Verificando se há estadia em andamento...");
1013   $scope.estadiaEmAndamento = false;
1014   $http.get($rootScope.url + '/estadia/estadiaAberta/?access_token=' +
1015     window.sessionStorage.getItem('token')).then(function (response) {
1016     if (response.data.idEstadia != null) {
1017       $scope.estadiaEmAndamento = true;
1018       $scope.estadia = response.data;
1019     }
1020
1021     console.log($scope.estadiaEmAndamento);
1022     if ($scope.estadiaEmAndamento == true) {
1023       var dataEntrada = new Date(response.data.dataEntrada);
1024       $scope.estadia.dataEntrada = dataEntrada;
1025       $scope.startTime = dataEntrada.getTime();
1026       $scope.timerConsole = '';
1027       $scope.timerType = '';
1028       $scope.$broadcast('timer-start');
1029       $scope.timerRunning = true;
1030       console.log("Encontrou estadia: " + $scope.estadia);
1031     }
1032     $scope.hideLoading();
1033   }).catch(function (response) {
1034     $scope.hideLoading();
1035     if (response.data.error == 'invalid_token') {
1036       console.log("token inválido.");
1037       $state.go("login");
1038     } else
1039       GlobalFunctions.showAlert('Erro', 'Não foi possível consultar estadias: ' + response.data.message);
1040   })

```

Fonte: elaborado pelo autor.

O Quadro 7 traz a função `verificaEstadiaAndamento`, que efetua um GET (linha 1014) no *back-end* para consultar a existência de uma estadia em andamento do usuário em questão. Após retorno do JavaScript Object Notation (JSON) da requisição, caso a tag `response.data.idEstadia` (linha 1016) seja diferente de nulo, considera-se existente uma estadia, e por conseguinte inicia-se a temporização do timer na linha 1025 com base na variável `dataEntrada` – data de entrada no estacionamento. Em contrapartida, constatada nulidade da tag `response.data.idEstadia`, é considerado que não existe estadia em andamento, por conseguinte, é disponibilizada na tela de estadias um botão (Quadro 6, linha 9) para permitir que o usuário possa iniciar uma estadia.

Já na camada *servidor* (*back-end*), demonstrado no Quadro 8, apresenta-se o endpoint utilizado na requisição (linha 1014, Quadro 7) efetuada no *front-end*. Para consultar se de fato existe estadia em andamento é executada a função `findAbertaUsuario` (Quadro 9). Desta forma, quando encontrada uma estadia em aberto, é retornado um JSON com os dados da estadia, possibilitando que sejam apresentadas as informações da mesma na camada *cliente*. Por outro lado, quando não encontrada uma estadia em aberto, o endpoint retorna uma JSON em branco.

Quadro 8 – Endpoint `estadiaAberta` do controlador de estadias: backend

```

44 @RequestMapping(produces = MediaType.APPLICATION_JSON_VALUE,
45                 method = RequestMethod.GET,
46                 value = "/estadiaAberta")
47 public ResponseEntity<EstadiaPojo> getEstadiaAbertaUsuario() {
48     return new ResponseEntity<>(estadiaDao.findAbertaUsuario(), HttpStatus.OK);
49 }

```

Fonte: elaborado pelo autor.

Quadro 9 - Função `findAbertaUsuario`: back-end

```

151 public EstadiaPojo findAbertaUsuario() {
152     System.out.println("Realizando consulta de estadia aberta.");
153
154     UsuarioEntity usuario = null;
155     DetachedCriteria criterioUsuario = DetachedCriteria.forClass(UsuarioEntity.class);
156     criterioUsuario.add(Restrictions.eq("login", SecurityContextHolder.getContext().getAuthentication().getName()));
157     List<UsuarioEntity> usuarioList = (List<UsuarioEntity>) hibernateTemplate.findByCriteria(criterioUsuario);
158
159     if (!usuarioList.isEmpty()) {
160         usuario = usuarioList.get(0);
161         System.out.println("Encontrou usuário. " + usuario.toString());
162     }
163
164     DetachedCriteria criterio = DetachedCriteria.forClass(EstadiaEntity.class);
165     criterio.createAlias("usuario", "usu");
166     criterio.add(Restrictions.eq("usu.id", usuario.getId()));
167     criterio.add(Restrictions.isNull("dataSaida"));
168     criterio.add(Restrictions.isNotNull("dataEntrada"));
169     List<EstadiaPojo> list = findAll(criterio);
170
171     if (!list.isEmpty()) {
172         return list.get(0);
173     } else
174         return new EstadiaPojo();

```

Fonte: elaborado pelo autor.

A função `findAbertaUsuario`, trazida no Quadro 9, pertencente a classe `EstadiaDao`, realiza inicialmente o carregamento do usuário (linha 157) pelo Hibernate, e posteriormente, na linha 169 consultar se há alguma estadia cujo `usuario.id` é correspondente ao `id` do

usuário da sessão ativa, com a `dataSaida` nula, e `dataEntrada` diferente de nulo. Por fim, o método retorna uma instância da classe `EstadiaPojo`.

Quando verificada inexistência de `estadia` em aberto, o usuário pode iniciar uma estadia, e para isto, primeiramente o método `scanNow`, disposto no Quadro 10, abre a câmera do dispositivo móvel para leitura do QR Code (seção 2.2) na linha 1056, por meio da função `$cordovaBarcodeScanner.scan`. Após feita a leitura, na linha 1060 é enviado a variável `idEstacionamento`, obtida pela leitura do QR Code, na execução do `POST` no endpoint `iniciarEstadia` (seção 2.2) do *back-end*. Logo que retornado, inicia-se a contagem do tempo decorrido por meio do *timer* na linha 1069.

Quadro 10 - Método do front-end para iniciar uma estadia

```

1053  $scope.scanNow = function() {
1054      console.log("iniciando estadia...");
1055
1056      $cordovaBarcodeScanner.scan().then(function(imageData) {
1057          var idEstacionamento = parseInt(imageData.text, idEstacionamento);
1058
1059          $scope.showLoading();
1060          $http.post($rootScope.url + '/estadia/iniciarEstadia/' + idEstacionamento
1061                  + '?access_token=' + window.sessionStorage.getItem('token')).then(function(response) {
1062              $scope.estadiaEmAndamento = true;
1063              $scope.estadia = response.data;
1064              $scope.estadia.dataEntrada = response.data.dataEntrada;
1065              var inicio = new Date(response.data.dataEntrada);
1066              $scope.startTime = new Date();
1067              $scope.timerConsole = '';
1068              $scope.timerType = '';
1069              $scope.$broadcast('timer-start');
1070              $scope.timerRunning = true;
1071              $scope.hideLoading();
1072              GlobalFunctions.showAlert('Aviso', 'Checkin confirmado!');
1073          }).catch(function(response) {
1074              $scope.hideLoading();
1075              GlobalFunctions.showAlert('Aviso', 'Não foi possível realizar checkin: ' + response.data.message);
1076          })
1077
1078      }, function(error) {
1079          console.log("Não foi possível realizar a leitura. Tente novamente!");
1080      });
1081  };

```

Fonte: elaborado pelo autor.

Por sua vez, no *back-end*, a fim de iniciar uma estadia, o Quadro 11 traz o endpoint responsável por iniciar as estadias dos usuários. O endpoint é requisitado por meio de um `POST` no *front-end*, que passa como parâmetro a variável `idEstacionamento`, carregada da leitura do QR Code por meio da câmera do dispositivo móvel do `Usuario`.

Quadro 11 - Endpoit para iniciar uma estadia no back-end

```

62  @RequestMapping(produces = MediaType.APPLICATION_JSON_VALUE,
63      method = RequestMethod.POST,
64      value = {"/iniciarEstadia/{idEstacionamento}}")
65  public ResponseEntity<EstadiaPojo> iniciarEstadia(@PathVariable("idEstacionamento") Long idEstacionamento) {
66      return new ResponseEntity<>(estadiaDao.iniciarEstadia(idEstacionamento), HttpStatus.OK);
67  }

```

Fonte: elaborado pelo autor.

Desta forma, é executada a função `iniciarEstadia` passando o parâmetro `idEstacionamento` para identificar em qual estacionamento está sendo iniciada a estadia, que, posteriormente, é utilizado para realizar uma busca no banco de dados a fim de obter os



dados do estacionamento (Quadro 12, linha 213). Já o método `iniciarEstadia`, apresentado no Quadro 12, instancia uma nova estadia na linha 212 por meio da classe `EstadiaPojo`. Logo após, na linha 213, busca no banco de dados, por meio do Hibernate, o estacionamento no qual está se iniciando uma estadia e atribui o resultado da busca à estadia em questão. Na linha 214 é atribuída a data de entrada no estacionamento, e, por fim, efetuada a gravação da estadia por meio da função `save` na linha 216. Desta forma, após execução e retorno da função `iniciarEstadia`, no *front-end* é mostrado ao usuário as informações da estadia em andamento. Por outro lado, ao finalizar uma estadia no *front-end*, é chamado o método `finalizarEstadia` do *back-end* por meio de uma requisição HTTP. O método `finalizarEstadia` é demonstrado no Quadro 13.

Quadro 12 - Função do back-end para iniciar estadia

```

208@ @Transactional(rollbackFor = Throwable.class)
209 public EstadiaPojo iniciarEstadia(Long idEstacionamento) {
210     System.out.println("Iniciando estadia no estacionamento id: "+ idEstacionamento);
211
212     EstadiaPojo estadia = new EstadiaPojo();
213     estadia.setEstacionamento(hibernateTemplate.get(EstacionamentoEntity.class, idEstacionamento));
214     estadia.setDataEntrada(new GregorianCalendar(locale).getTime());
215
216     return this.save(estadia, null);
217 }

```

Fonte: elaborado pelo autor.

Quadro 13 - Função do back-end para finalizar uma estadia

```

178@ @Transactional(rollbackFor = Throwable.class)
179 public EstadiaPojo finalizarEstadia(Long idEstacionamento) {
180     EstadiaPojo estadia = findAbertaUsuario();
181     EstacionamentoEntity estacionamento = hibernateTemplate.load(EstacionamentoEntity.class, idEstac);
182     Date dataSaida = new GregorianCalendar(locale).getTime();
183     estadia.setDataSaida(dataSaida);
184     Instant inicio = null;
185     inicio = estadia.getDataEntrada().toInstant();
186     Instant fim = dataSaida.toInstant();
187     Duration d = Duration.between(inicio, fim);
188     Long tempo = d.toMinutes();
189     estadia.setPreco(estacionamento.getPreco());
190
191     if (tempo > 60) {
192         long horas = (tempo / 60);
193         double resto = Math.floorMod(tempo, 60);
194         if (resto == 0)
195             estadia.setPreco(estacionamento.getPreco() * horas);
196         else
197             estadia.setPreco((estacionamento.getPreco() * horas) + estacionamento.getPreco());
198     }
199
200     String idPagamento = realizarPagamento(estadia);
201     System.out.println("Pagou - ID: " + idPagamento);
202     estadia.setIdPagamento(idPagamento);
203
204     return save(estadia, estadia.getIdEstadia());
205 }

```

Fonte: elaborado pelo autor.

No Quadro 13, após carregar a estadia por meio do método `findAbertaUsuario` (linha 180), define a data de saída (linha 183) da estadia e calcula, por meio da função `Duration.between`, na linha 187, o tempo entre a data de entrada e saída, encontrando assim, o tempo total da estadia ao converte-lo para minutos na linha 188. Em seguida, da linha 191 até 198, é calculado o preço a ser pago pela estadia. Por fim, é invocado o método



realizarPagamento para efetivar o pagamento da estadia junto ao ambiente de testes da Cielo Sandbox. O Quadro 14 traz o método realizarPagamento do *back-end*, demonstrando a maneira de utilizar a API da Cielo.

Quadro 14 - Função do back-end para realizar o pagamento da estadia

```

219 public String realizarPagamento(EstadiaPojo estadia) {
220     System.out.println("Iniciando pagamento via cartão de crédito!");
221     Merchant merchant = new Merchant(this.MERCHANT_ID, this.MERCHANT_KEY);
222
223     // Cria uma instância de Sale informando o ID do pagamento
224     Sale sale = new Sale(estadia.getIdEstadia().toString());
225
226     // Cria uma instância de Payment informando o valor do pagamento
227     int valor = new Integer( Double.toString(Math.round(estadia.getPreco() * 100)).replace(".0", ""));
228     Payment payment = sale.payment(valor);
229
230     UsuarioEntity usuario = null;
231     DetachedCriteria criteriaUsuario = DetachedCriteria.forClass(UsuarioEntity.class);
232     criteriaUsuario.add(Restrictions.eq("login", SecurityContextHolder.getContext().getName()));
233     List<UsuarioEntity> usuarioList = (List<UsuarioEntity>) hibernateTemplate.findByCriteria(criteriaUsuario);
234
235     if (!usuarioList.isEmpty()) { usuario = usuarioList.get(0); }
236
237     DetachedCriteria criteria = DetachedCriteria.forClass(FormaPagamentoEntity.class);
238     criteria.createAlias("usuario", "usu");
239     criteria.add(Restrictions.eq("usu.id", usuario.getId()));
240     List<FormaPagamentoEntity> list = (List<FormaPagamentoEntity>) hibernateTemplate.findByCriteria(criteria);
241     FormaPagamentoEntity cartao = list.get(0);
242     // Cria uma instância de Credit Card utilizando os dados de teste
243     payment.creditCard(cartao.getCodigoSeguranca(), cartao.getBandeira().setExpirationDate(cartao.getValidade())
244         .setCardNumber(cartao.getNumero().toString())
245         .setHolder(cartao.getNomeCartao()));
246
247     sale.setPayment(payment);
248     try {
249         // Configura o SDK com seu merchant e o ambiente apropriado para criar a venda
250         sale = new CieloEcommerce(merchant, Environment.SANDBOX).createSale(sale);
251         System.out.println("Pagamento confirmado! Retorno : " + sale.getPayment().toString());
252     } catch (CieloRequestException e) {
253         CieloError error = e.getError();
254         System.out.println("Deu ruim!" + error.getMessage());
255     } catch (IOException e) {
256         System.out.println("Deu ruim!" + e.getMessage());
257         e.printStackTrace();
258     }
259     return sale.getPayment().getPaymentId();

```

Fonte: elaborado pelo autor.

No Quadro 14, na linha 221, inicialmente é definida a identificação do merchant (comerciante), instanciando um objeto merchant e passando-o o MERCHANT\_ID e MERCHANT\_KEY – credenciais necessárias para utilizar os métodos da API, obtidas após realizar o cadastro na Sandbox Register<sup>12</sup>. Em seguida, é instanciado o objeto sale (linha 223), cujo qual leva para identificação o atributo idEstadia da classe Estadia. Ainda, a fim de determinar o valor a ser pago e o meio de pagamento que será utilizado, na linha 227 é agregado no objeto payment o valor da estadia – o qual deve ser formatado com ponto para determinar as casas decimais –, e na linha 242, o cartão de crédito para realizar o pagamento. Por fim, na linha 246, o objeto sale (venda) recebe o meio de pagamento anteriormente definido, e, efetiva o envio à plataforma Cielo Sandbox<sup>13</sup> ao executar o método createSale na linha 249. Desta forma, após enviada a requisição, a

<sup>12</sup> Disponível em: <<https://cadastrosandbox.cieloecommerce.cielo.com.br/>>. Acesso em: 16 jul. 2018.

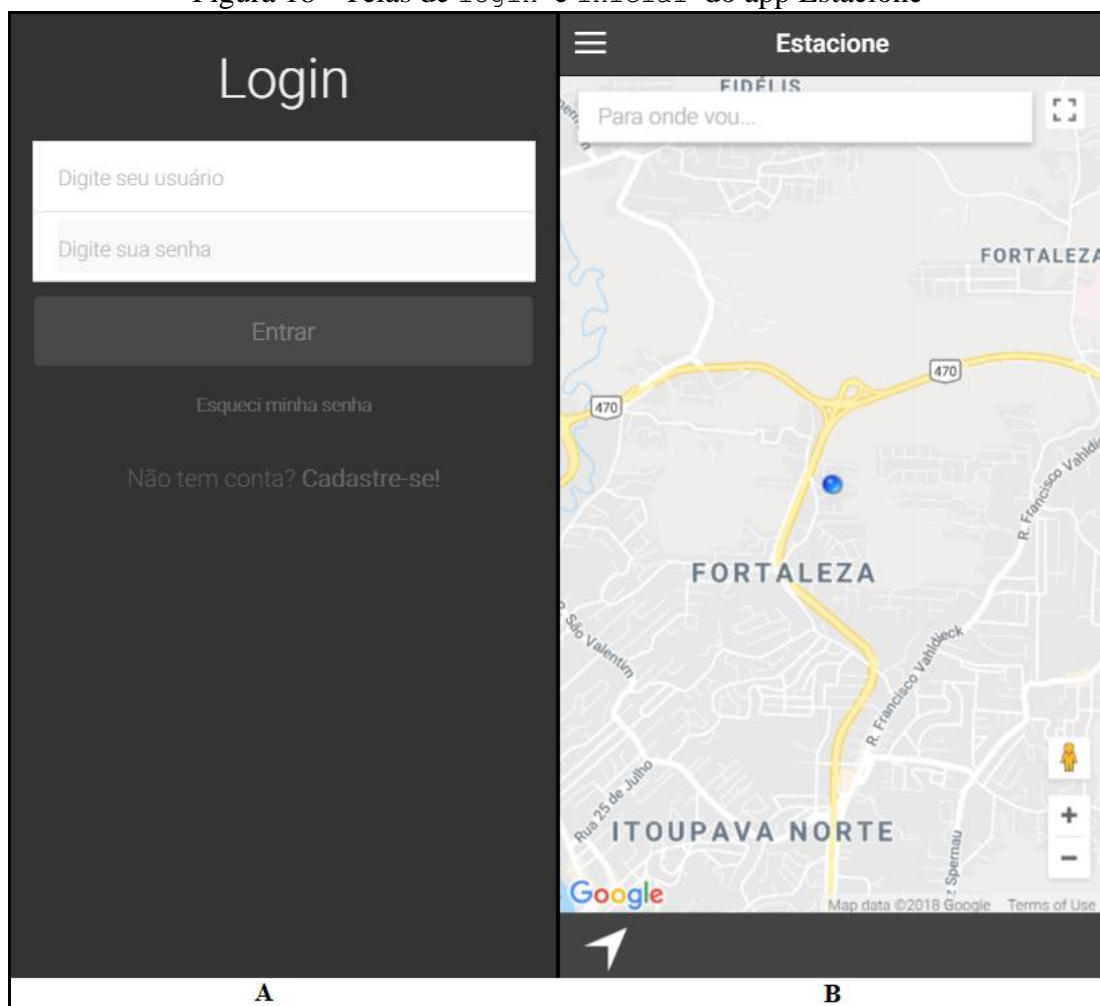
<sup>13</sup> Disponível em: <<https://developercielo.github.io/manual/cielo-e-commerce#sandbox-e-ferramentas>>. Acesso em: 16 jul. 2018.

plataforma retorna um `id`, que identifica o pagamento realizado, e, do qual, posteriormente, pode-se utilizar a fim de consultar o pagamento diretamente nos serviços disponibilizados pela API.

### 3.3.3 Operacionalidade da Implementação

Nesta subseção será demonstrado o funcionamento do app Estacione desenvolvido mediante as telas produzidas na implementação. Primeiramente, o app Estacione apresenta a tela de login (Figura 18A), na qual o usuário preenche os dados de sua conta e pressiona o botão `entrar` para ingressar no app, ou, não possuindo uma conta, permite-o registra-la. Após efetuado o login, o usuário é redirecionado à tela inicial do app Estacione (Figura 18B), que apresenta um mapa com sua localização por meio de um marcador azul no centro do mapa. Além disso, também são carregados os estacionamentos no mapa.

Figura 18 - Telas de login e inicial do app Estacione



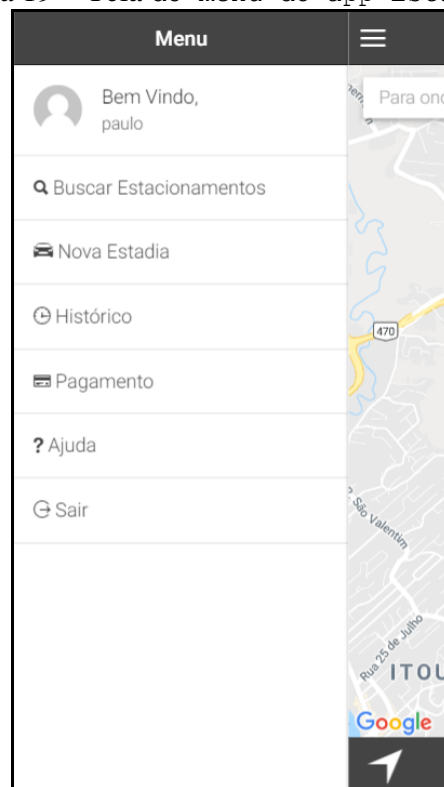
Fonte: elaborado pelo autor.

O menu do app Estacione, apresentado na Figura 19, é aberto quando pressionado o botão identificado com as três linhas horizontais, no canto superior direito da Figura

18B, ou, no momento que o usuário arrasta, da esquerda para a direita, o dedo sobre a tela do aplicativo. Desta forma, o aplicativo dispõe ao usuário as seguintes funcionalidades:

- a) buscar estacionamentos: direciona o usuário para a tela principal;
- b) nova estadia: direciona o usuário para a tela de estadias;
- c) histórico: direciona o usuário para a tela que exibe o histórico de estadias;
- d) pagamento: direciona o usuário para o cadastro da forma de pagamento;
- e) ajuda: direciona o usuário para tela das opções de ajuda do app Estacione;
- f) sair: efetua o *logout* do usuário do app Estacione.

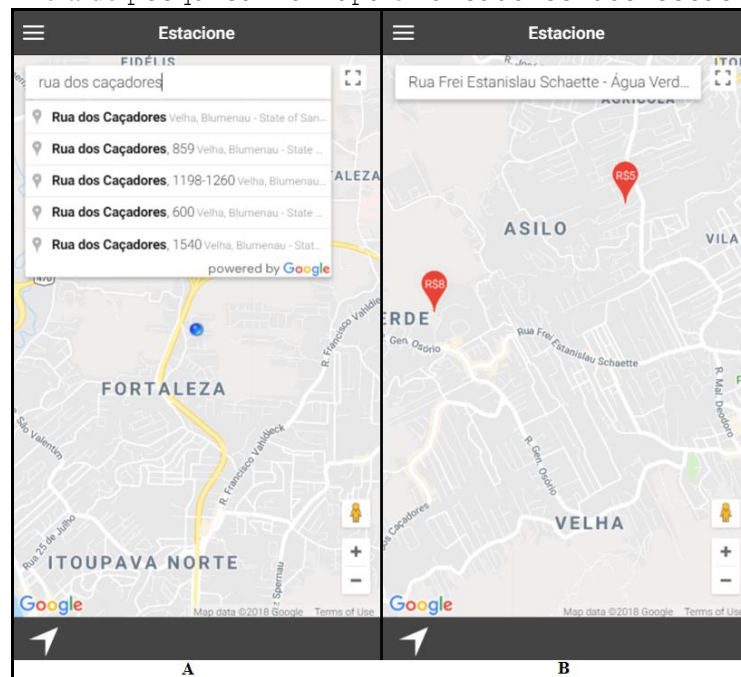
Figura 19 – Tela do menu do app Estacione



Fonte: elaborado pelo autor.

No mapa é possível pesquisar por locais de referência e endereços, conforme demonstrado na Figura 20A. Após selecionar o local desejado, o mapa posiciona a tela acerca do local e possibilita visualizar os estacionamentos nas proximidades, dispondo do valor praticado pelo estacionamento (Figura 20B). Além disso, quando pressionado o marcador no mapa, é aberta uma caixa de informação listando informações sobre o estacionamento, a qual apresenta o nome do estabelecimento, o preço por hora na estadia, e ainda, um hiperlink para abrir no Google Maps nativo do dispositivo móvel.

Figura 20 – Tela de pesquisa no mapa e marcadores dos estacionamentos



Fonte: elaborado pelo autor.

Primordialmente, o usuário precisa cadastrar seu cartão de crédito como forma de pagamento no aplicativo. A Figura 21 retrata a tela para essa finalidade, na qual obrigatoriamente, o usuário deve informar o número do cartão de crédito, o nome do impresso no cartão – normalmente, o proprietário do cartão –, a validade – composta por mês e ano –, e ainda, o código de segurança do cartão do credito. Nesta tela, o botão gravar é habilitado somente se todos os campos estiverem preenchidos.

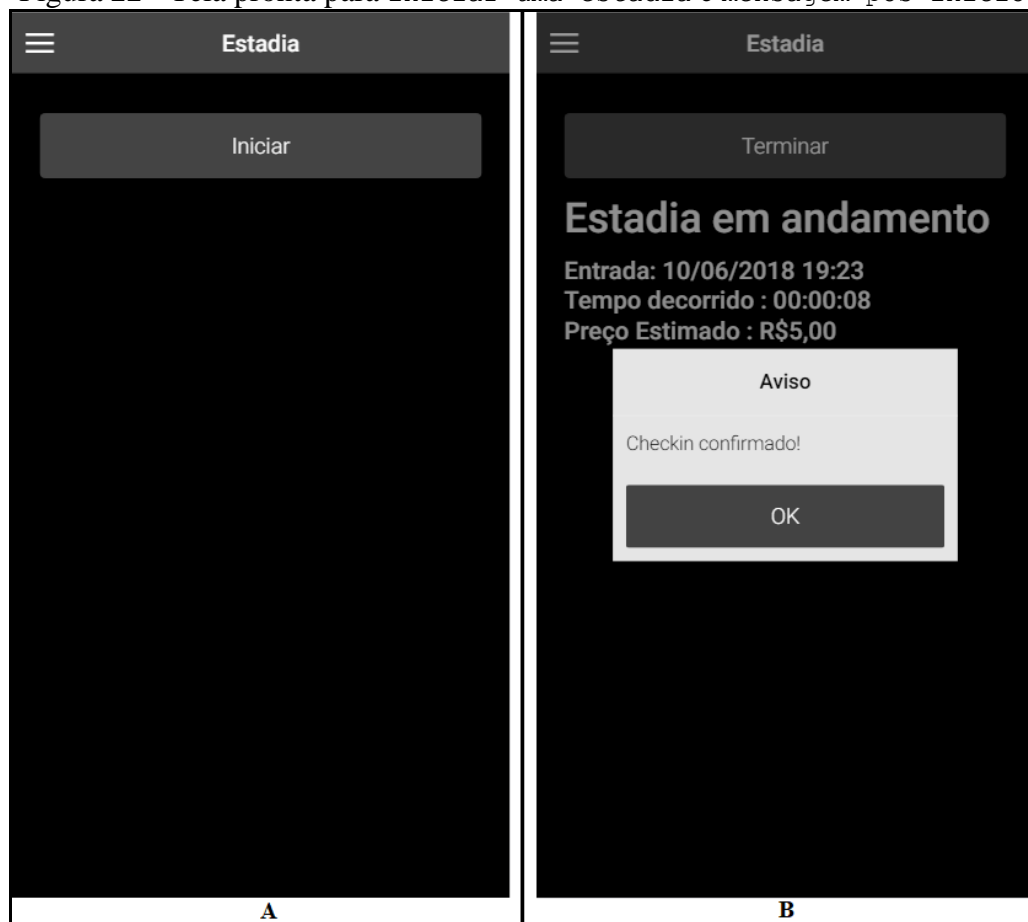
Figura 21 - Tela para cadastro da forma de pagamento

The image shows a vertical form titled 'Forma de Pagamento'. The form is divided into sections:  
 - 'Identificação' section: contains two input fields, 'Número do cartão' and 'Nome impresso no cartão'.  
 - 'Validade' section: contains two input fields, 'Mês' and 'Ano'.  
 - 'DVV' section: contains one input field, 'Código de segurança'.  
 - At the bottom of the form is a dark grey button labeled 'Gravar'.  
 The form has a dark background with white text and input fields.

Fonte: elaborado pelo autor.

A Figura 22 apresenta a principal tela do aplicativo, a qual incumbe a função de registrar a entrada e saída do estacionamento. Na Figura 22A, a tela mostra um botão com a opção disponível no momento – podendo ser para iniciar ou terminar uma estadia. Para que se inicie ou termine uma estadia, precisa-se identificar em qual estacionamento o usuário se encontra, para isso, armazena-se o id do estacionamento no QR Code, que na prática, seria impresso em um painel físico e pendurado em local de fácil acesso dentro do estacionamento. Ao pressionar o botão *Iniciar* é acionada a câmera do dispositivo móvel para que o usuário a posicione frente ao código de barras QR Code e realize a leitura do mesmo. Após feita a leitura, é exibida a mensagem apresentada na Figura 22B, que significa, neste caso, que o *back-end* iniciou com sucesso a estadia no estacionamento. Em qualquer outra situação adversa, se exibe a mesma caixa de mensagem, porém, com a seguinte mensagem, seguida do erro retornado do servidor: Não foi possível realizar checkin.

Figura 22 - Tela pronta para iniciar uma estadia e mensagem pós início



Fonte: elaborado pelo autor.

Depois de iniciada uma estadia, sempre que o usuário acessar a tela de estadia serão dispostas a data de entrada, o tempo decorrido e o preço estimado,

conforme apresentado na Figura 23A. A qualquer momento o usuário pode terminar uma estadia pressionando o botão Terminar. Ao pressioná-lo, imediatamente é acionada a câmera do dispositivo, da mesma forma que no ato de iniciar uma estadia, e, ao ler o QR Code é enviado ao *back-end* para efetivar o término da estadia. Por fim, o *back-end* envia o valor pago à plataforma da Cielo para efetivação do pagamento e retorna ao *front-end* para que se exiba uma mensagem avisando que o checkout foi confirmado, conforme demonstra a Figura 23B.

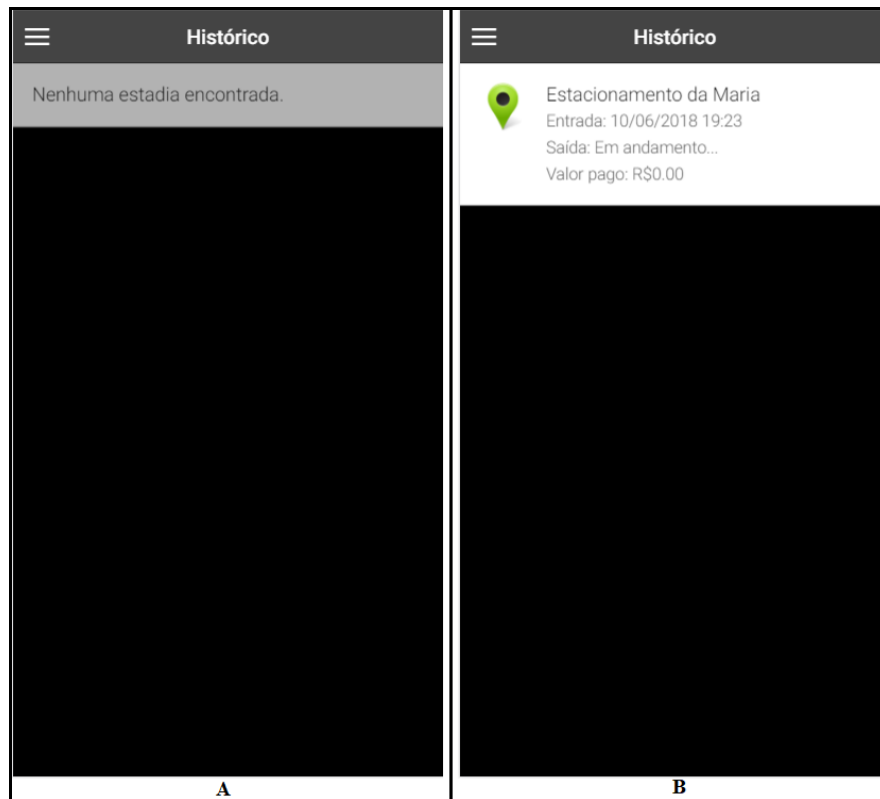
Figura 23 - Tela de estadia com estadia em andamento e pós término de estadia



Fonte: elaborado pelo autor.

Para fins de controle, o aplicativo dispõe de uma tela para consulta do histórico de estadias realizadas. Ao acessar a tela, quando inexistente estadia previamente realizada pelo usuário, apresenta-se uma mensagem fixa, conforme demonstra a Figura 24A, com o seguinte aviso: Nenhuma estadia encontrada. Em contrapartida, a Figura 24B mostra a tela quando encontradas estadias, situação na qual exibe-se uma lista contendo informações como: nome do estacionamento no qual se estacionou, data de entrada e saída do estacionamento, e ainda, o valor pago na estadia.

Figura 24 - Tela de histórico sem nenhuma estadia encontrada e com estadia encontrada



Fonte: elaborado pelo autor.

### 3.4 RESULTADOS E DISCUSSÕES

Nesta seção é realizada a comparação entre os trabalhos correlatos e o app Estacione desenvolvido (subseção 3.4.1), assim como é detalhado o processo de avaliação de usabilidade utilizando junto aos seus utilizadores. Ademais, são realizadas as discussões e apresentados os resultados obtidos durante o processo de busca da solução (subseção 3.4.2).

#### 3.4.1 Comparação entre o app Estacione desenvolvido e os trabalhos correlatos

Na presente subseção se realiza uma comparação entre os trabalhos correlatos apresentados na seção 2.4 e o app Estacione desenvolvido. O Quadro 15 traz algumas características dos trabalhos correlatos e uma comparação perante o app Estacione desenvolvido. Notoriamente, percebe-se que o aplicativo ParkMe (INRIX INC., 2018) contempla maior parte das características. Observa-se pelo Quadro 15, que as características disponibiliza informação sobre os estacionamentos mais próximos e mais baratos e permite pesquisar por estacionamento se destacam pelo fato de estarem contempladas nos correlatos Ferreira (2014), Vaga Inteligente (ESTAPAR, 2018), ParkMe (INRIX INC., 2018) e no aplicativo desenvolvido neste trabalho, e depende-se disso a importância dessa característica

pelo fato de possibilitar a busca por estacionamentos e a escolha pelo mais próximo do motorista, além de permitir a comparação dos valores praticados para auxiliar na sua decisão.

Quadro 15 - Comparativo dos correlatos perante o trabalho desenvolvido

Características \ Correlatos	Ferreira (2014)	Vaga Inteligente Estapar (2018)	ParkMe Inrix Inc. (2018)	Estacione (2018)
Plataforma	Android	Android e IOS	Android, IOS e Windows Phone	Android e IOS
Permite pesquisar por estacionamentos.	✓	✓	✓	✓
Disponibiliza informação sobre os estacionamentos mais próximos e mais baratos.	✓	✓	✓	✓
Permite visualizar informações sobre estacionamento livre na rua.	✓	X	✓	X
Permite visualizar informações sobre estacionamentos privados.	X	✓	✓	✓
Apresenta disponibilidade do lugar de estacionamento em tempo real.	X	✓	✓	X
Apresenta estacionamentos de qualquer cidade.	X	✓	✓	✓
Permite reservar vaga no estacionamento.	X	✓	✓	X
Utilizar pagamento via aplicativo móvel.	X	✓	✓	✓
Realiza a marcação de entrada e saída do estacionamento via QR Code	X	X	X	✓
Apresenta avaliação de usabilidade	X	X	X	✓

Fonte: elaborado pelo autor.

Os aplicativos de Ferreira (2014) e ParkMe de Inrix Inc (2018) possuem a característica permite visualizar informações sobre estacionamento livre na rua, a qual não foi abordada pelo aplicativo Estacione (2018). Em contrapartida, o aplicativo desenvolvido, juntamente o Vaga Inteligente (ESTAPAR, 2018) e ParkMe (INRIX INC. 2018), permitem a visualização de estacionamentos privados conforme observa-se na característica permite visualizar informações sobre estacionamentos privados.

No Quadro 15 é possível perceber que todos os aplicativos móveis de destacam por disponibilizarem seus aplicativos para a plataforma Android. Os apps Vaga Inteligente (ESTAPAR, 2018), ParkMe (INRIX INC., 2018) e o Estacione (2018) também se destacam devido à compatibilidade com às plataformas iOS. O app ParkMe (INRIX INC., 2018) destaca-se ainda por disponibilizar seu app para a plataforma Windows Phone. Observa-se também que todos os aplicativos móveis, exceto Ferreira (2014), apresentam os estacionamentos de qualquer cidade e utilizam o pagamento móvel no aplicativo, conforme demonstram as características Apresenta estacionamentos de qualquer cidade e Utiliza pagamento via aplicativo móvel, respectivamente.



Pode-se observar ainda que os aplicativos móveis Vaga Inteligente (ESTAPAR, 2018) e ParkMe (INRIX INC., 2018) apresentam, em tempo real, informação sobre a disponibilidade de vagas do estacionamento. Além disso, ambos os aplicativos permitem que os usuários reservem vagas de estacionamento, tornando o app bastante útil em situações que se realiza uma compra de um ingresso para um show e necessita-se garantir uma vaga antecipadamente.

O aplicativo Estacione (2018) possui maior semelhança aos aplicativos ParkMe (INRIX INC., 2018) e Vaga inteligente (ESTAPAR, 2018), principalmente por permitir o pagamento da estadia em estacionamentos privados por meio do aplicativo móvel. Assemelha-se também por auxiliar o usuário a encontrar o estacionamento financeiramente mais acessível e próximo do local desejado, característica essa, que se apresenta no protótipo de Ferreira (2014).

Por fim, percebe-se que nenhum dos correlatos permite realizar a marcação de entrada e saída do estacionamento, bem como não apresentam uma avaliação de usabilidade. Neste sentido, observa-se um diferencial no app Estacione (2018), que além de ter utilizado o QR Code para marcar as entradas e saídas do estacionamento, buscou avaliar o aplicativo por meio da avaliação de usabilidade e experiência do usuário, focando em proporcionar uma utilização mais confortável ao usuário.

### 3.4.2 Avaliação de usabilidade e experiência de usuário

Para validar a eficiência e verificar a usabilidade do app Estacione desenvolvido, foi realizado uma avaliação de usabilidade e experiência de usuário (seção 2.3), tendo como base a experiência do usuário em sua utilização. A avaliação foi realizada por meio de questionário de avaliação de forma quantitativa e qualitativa, de forma online; elaborado por meio do Google Formulários<sup>14</sup> e da Ferramenta Mentimeter<sup>15</sup>. A avaliação foi disponibilizada para os participantes da pesquisa por meio de um aplicativo de mensagens móveis, contendo um link para acesso a uma página do Google Formulários. Foi utilizada uma amostragem de 15 usuários participantes selecionados de forma aleatória para o uso do app Estacione. A avaliação de usabilidade e experiência de usuário foi dividida em duas partes: a primeira parte diz respeito a conhecer melhor os participantes da avaliação; e a segunda parte visou a avaliação em si utilizando partes do método M3C-URUCAg (seção 2.3). Os usuários

---

<sup>14</sup> Disponível em: <<https://docs.google.com/forms/>>. Acesso em: 18 jun. 2018.

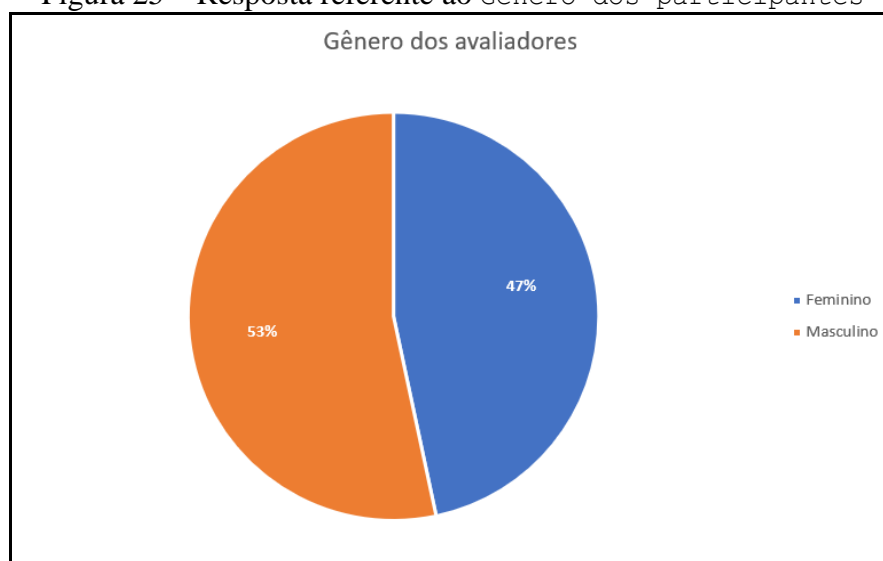
<sup>15</sup> Disponível em: <<http://www.mentimeter.com>>. Acesso em: 18 jun. 2018.

participantes que fizeram parte da amostra tiveram conhecimento do app Estacione por meio da apresentação presencial no computador do autor.

A autora do método, Costa (2018), defende a existência de outras realidades, além daquelas que os especialistas de um aplicativo são oriundos do mercado ou da academia, nas quais o especialista do aplicativo acaba sendo o próprio usuário final – por possuir a expertise necessária –, como é o caso do app Estacione. Sendo assim, de modo que os usuários participantes obtivessem um melhor entendimento quanto ao app Estacione e a avaliação de usabilidade, foi disponibilizado um roteiro de tarefas, cujo objetivo era auxiliar os participantes no uso do aplicativo (Apêndice C).

Os usuários participantes tiveram acesso por um link a uma página do Google Formulários, contendo: o Termo de Consentimento Livre e Esclarecido (TCLE) (Apêndice D), que convidava o usuário participante a colaborar com a pesquisa, informando sobre os possíveis danos e os direitos ao participar da avaliação. Além disso, no final do TCLE havia uma orientação ao usuário participante, informando-o que caso prosseguisse para a seção do roteiro (Apêndice C) e da avaliação, suas informações não seriam divulgadas. A primeira parte do questionário é referente a conhecer melhor os participantes da pesquisa. As três primeiras perguntas são direcionadas ao perfil dos participantes, buscando compreender gênero, faixa etária e área de atuação profissional. A respeito do perfil dos participantes, as informações obtidas na aplicação da avaliação de usabilidade e UX são exibidas na Figura 25, Figura 26 e Figura 27.

Figura 25 – Resposta referente ao Gênero dos participantes



Fonte: elaborado pelo autor.

Conforme demonstra a Figura 25, buscou-se obter uma homogeneidade na quantidade de avaliadores por gênero. A pesquisa foi respondida por participantes do sexo feminino e

masculino, resultando em 47% da avaliação realizada por participantes do gênero feminino e, com uma pequena superioridade, 53% por participantes do gênero masculino (Figura 25). Desta forma, objetou-se alcançar um equilíbrio, evitando-se assim, qualquer favorecimento perante resultado e o viés das pesquisas.

Ademais, a avaliação também buscou saber a idade deles. Neste quesito, na Figura 26 observa-se uma variação maior entre os envolvidos. A faixa etária predominante está entre vinte e três e vinte e sete anos, com 33%. Os indivíduos maiores de cinquenta anos totalizam 27%, seguidos de 20% entre trinta e três e trinta e sete anos. Em contrapartida, somente 13% totalizam a faixa entre dezoito e vinte e dois anos e, por fim, o menor público observado na pesquisa, distribuído entre trinta e oito a quarente e dois anos, totalizando 7% dos avaliadores (Figura 26).

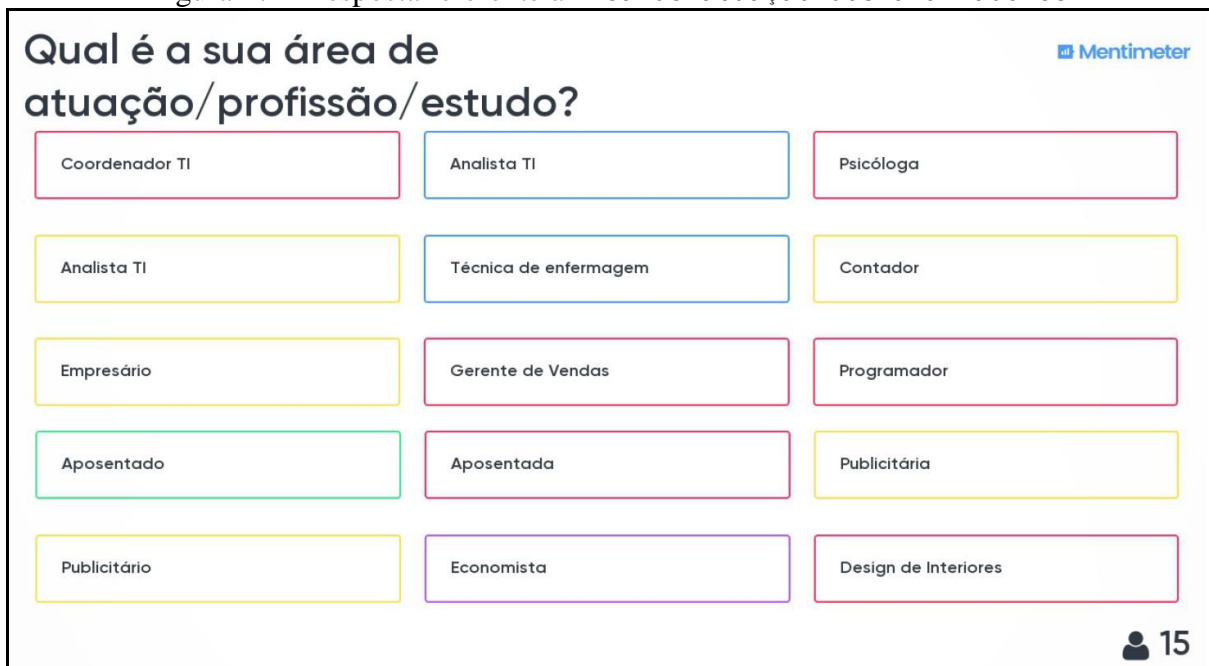
Figura 26 – Resposta referente a Faixa etária dos participantes



Fonte: elaborado pelo autor.

Com relação à área de atuação, a amostra apresentada na Figura 27 demonstra predominância da área de tecnologia entre os participantes, totalizando 26,67% (n=4), sendo: um coordenador de TI, dois analistas de TI e um programador. Seguidos 13,33% (n=2) de aposentados e 13,33% (n=2) de publicitários. Por fim, os demais participantes totalizam aproximadamente 50% (n=7) da pesquisa, divididos nas áreas de: psicologia; enfermagem; contabilidade; economia; design de interiores e empresarial. Desta forma, observa-se certa heterogeneidade dentre os participantes da pesquisa, possibilitando obter uma maior abrangência no que diz respeito à percepção perante o aplicativo.

Figura 27 – Resposta referente à Área de atuação dos avaliadores



Fonte: elaborado pelo autor.

A segunda parte do questionário diz respeito a avaliação de usabilidade e experiência do usuário, composta por 17 perguntas, dentre estas, 13 perguntas objetivas e 4 perguntas descritivas sobre os pontos positivos e negativos do sistema, bem como sugestões de melhorias. O objetivo das perguntas de usabilidade do app Estacione foi identificar a satisfação do usuário em relação a utilização dos mecanismos fornecidos pelo app. As perguntas foram elaboradas pelo método, que utiliza os fundamentos das heurísticas de Nielsen, dispostas de forma resumida no Quadro 3 (seção 2.3) e no Quadro 16 a relação das perguntas utilizadas na avaliação com as heurísticas.

Quadro 16 – Relação das Heurísticas de Nielsen x Perguntas

Heurísticas	Perguntas do Questionário
H1	É possível identificar quando o aplicativo está carregando alguma informação, sem que pareça travado?
H2	O aplicativo utiliza palavras, termos, expressões e conceitos familiares para você?
	Quando em andamento, é facilmente identificado o tempo decorrido e o preço a pagar na estadia?
H3	O aplicativo possibilita que você saia da opção selecionada a qualquer momento?
H4	No histórico de estadias as informações sobre as estadias são claras?
	O aplicativo segue os padrões e estilos consistentes nas telas?
H5	O aplicativo deixa claro através de seus ícones o objetivo da opção?
H6	É possível memorizar onde encontrar uma opção anteriormente utilizada?
H7	O aplicativo possui teclas, imagens ou ícones para facilitar o seu uso?
H8	As mensagens do aplicativo trazem informações relevantes ou na maioria das vezes necessárias?
H9	O aplicativo possui mensagens de erros com linguagem simples?
H10	Através da ajuda do aplicativo é possível entender o funcionamento do aplicativo?

Fonte: elaborado pelo autor.

Na coleta dos resultados, o questionário de avaliação foi aplicado em seguida ao usuário ter utilizado o app Estacione, abordado na seção 2.3, bem como fazer uso de *emoticons* (Apêndice E) em conjunto com respostas pares (de um a quatro ou de um a seis) na Escala Likert, ao invés de verificar se a heurística foi violada ou não violada. A não inclusão de uma resposta neutra conforme sugerido pelo método, diz respeito aos usuários participantes terem de se posicionar de forma positiva ou negativa, a fim de obter o máximo de feedback e sugestões de melhorias dos usuários.

O grau de severidade (Quadro 2 apresentado na seção 2.3) pode ser obtido pelo feedback dos especialistas ou realizado por meio de pergunta no questionário. Essas considerações são realizadas para ajudar a evitar o viés de pesquisas. Assim, foram definidas quatro diferentes respostas com base na Escala Likert e *emoticons* representando cada uma das respostas, sendo elas: concordo totalmente, concordo parcialmente, discordo parcialmente e discordo totalmente (Apêndice E).

Os objetivos das perguntas de usabilidade e experiência do usuário no app Estacione foram para identificar problemas e o seu grau de gravidade, como: baixa (1), média (2), alta (3) e altíssima (4) (Quadro 2 apresentado na seção 2.3). A avaliação encontrou quatro problemas, conforme pode ser visualizado pelo Quadro 17. Para encontrar os problemas, foram filtrados os feedbacks negativos dos participantes. Das 13 perguntas que foram realizadas com os participantes apenas 4 delas tiveram respostas parcialmente negativas, representando gravidades 1. Não foi encontrada nenhuma resposta totalmente negativa e problemas de gravidade 2, 3 e 4, que impactariam diretamente na experiência e satisfação do usuário.

Quadro 17 – Relação das Heurísticas e problemas encontrados

<b>Heurística</b>	<b>Problema</b>	<b>Gravidade</b>
H1	--	--
H2	1	1
H3	--	--
H4	--	--
H5	1	1
H6	--	--
H7	--	--
H8	--	--
H9	1	1
H10	1	1

Fonte: elaborado pelo autor.

Observa-se no Quadro 17 que o participante problema encontrou um problema na heurística H2, cuja consiste em avaliar a correspondência entre o sistema e o mundo

real, tal resultado remete à alguma falha, perante a percepção do avaliador, nas expressões ou terminologias adotadas nas interfaces do aplicativo. Já o segundo problema encontrado foi na heurística que avalia o sistema perante a prevenção de erro (H5), problema do qual pode-se inferir que o avaliador encontrou algo que lhe induziu ao erro. Além disso, apresenta um problema na heurística H9, cuja consiste em avaliar o design estético e minimalista. Neste sentido, pressupõe-se que o problema da H9 foi causado, possivelmente, por alguma mensagem de erro inesperada, de modo que não deixou claro ao avaliar qual o problema ocorrido na ação realizada. Por fim, encontrou-se ainda um problema na opção de ajuda do aplicativo, observado por meio do resultado da heurística H10, que buscou entender eficácia ajuda e documentação do app Estacione.

A seguir são apresentados os resultados obtidos referente a avaliação de usabilidade e experiência. A avaliação busca compreender a forma como as informações são dispostas no app Estacione, sua relevância e o cumprimento dos objetivos colocados neste trabalho de conclusão de curso, bem como a facilidade do seu uso. No que se refere à visibilidade do estado do sistema (H1), 67% dos avaliadores notam que o aplicativo não apresenta nenhum sinal de travamento durante o uso e, apenas 33%, concordam parcialmente com esta observação. Em todas as telas do aplicativo apresentou-se um ícone central de loading ao abrir ou realizar algum processamento. Sendo assim, justifica-se o resultado obtido na pergunta 1 da Figura 28 e, no mínimo, remete ao fato que utilizar esse tipo de recurso é recomendável para demonstrar confiabilidade e feedback ao usuário.

Figura 28 - Resultado da pergunta 1

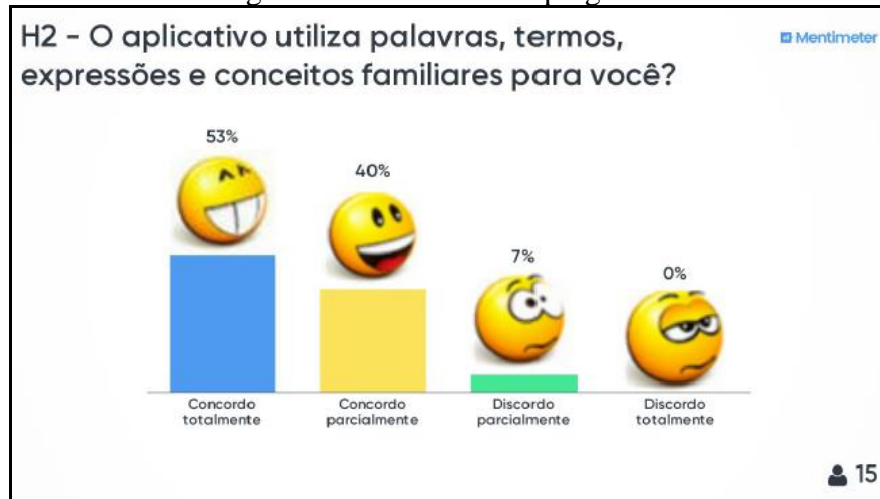


Fonte: elaborado pelo autor.

Com relação à correspondência entre o sistema e o mundo real (H2), a Figura 29 indica que 67% dos avaliadores se familiarizaram com os termos utilizados no aplicativo e se sentem confortáveis com as palavras dispostas, mostrando que as funcionalidades foram implementadas de forma simples e compreensível. Entretanto, 7% dos

participantes discordam parcialmente desta afirmação, fato do qual depreende-se que algum conceito, vocabulário ou processo não foi totalmente compreendido. Vale ainda ressaltar, que a faixa etária, área de atuação e a adaptação destes avaliadores com a tecnologia pode ter impactado no respotado desta pergunta.

Figura 29 - Resultado da pergunta 2



Ainda no que diz respeito à correspondência entre o sistema e o mundo real (H2), buscou-se questionar os participantes acerca de uma das principais funcionalidades do aplicativo desenvolvido: visualizar o tempo decorrido e o preço a pagar. A Figura 30 demonstra que 93% dos participantes concordaram totalmente que o app permite identificar o tempo decorrido e o preço a pagar na estadia, enquanto apenas 7% concordaram parcialmente. Sendo assim, conclui-se que o aplicativo concretizou o propósito objetivado na seção 1.1 (d), quanto à informar o tempo e o custo de uma estadia onde o veículo está estacionado.

Figura 30 - Resultado da pergunta 3



O controle e liberdade do usuário (H3), é referente a fornecer alternativas e possibilidades de refazer a operação, foi avaliado positivamente de acordo com o demonstrado na Figura 31. Dentro os participantes, 67% concordam totalmente que o aplicativo possibilita sair de uma opção selecionada. A opção cancelar foi disponibilizada em praticamente todas as telas do aplicativo, tal que, explica a satisfação dos usuários com relação a esta opção. Entretanto, há ainda 33% que concordam parcialmente (Figura 31), resultado cuja consequência se deve à falta de uma opção para abortar alguns processamentos nas telas, como por exemplo na abertura do mapa, que obriga o usuário a esperar até que sejam carregados todos os marcadores. Neste sentido, pode-se inferir que alguns avaliadores, possivelmente, consideraram relevante a ausência de uma opção para abortar os processamentos nas telas em sua avaliação.

Figura 31 - Resultado da pergunta 4



Fonte: elaborado pelo autor.

Com relação à consistência e padronização (H4), foi perguntado aos participantes: no histórico de estadias as informações sobre as estadias são claras? (Figura 32). O resultado apontou que 53% dos participantes concordam parcialmente que as informações apresentadas no histórico estão dispostas seguindo certa padronização, bem como 47% concordam totalmente com esta afirmação. Embora a maioria concorde com essa heurística, poder-se-ia ter listado informações sobre a efetivação do pagamento.



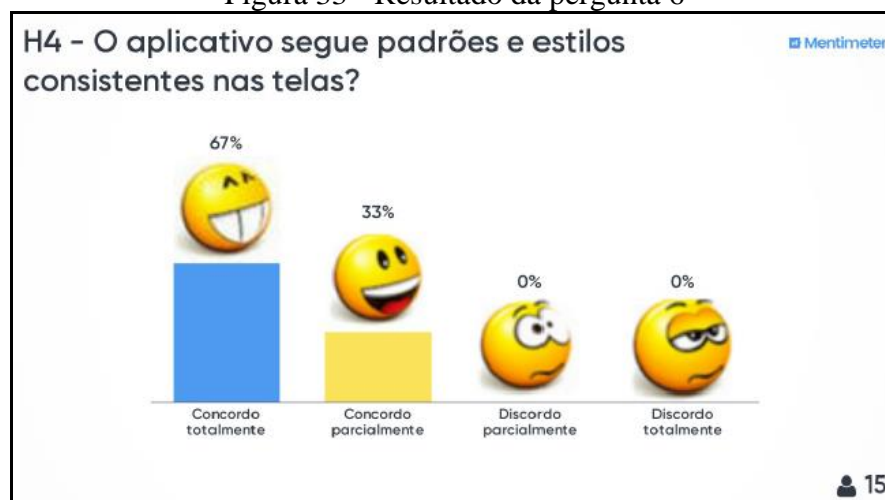
Figura 32 - Resultado da pergunta 5



Fonte: elaborado pelo autor.

A pergunta o aplicativo segue padrões e estilos consistentes nas telas? (Figura 33), também relacionada à consistência e padronização (H4), foi apresentada aos participantes a fim de avaliar sua percepção perante a interface do app Estacione. De acordo com o resultado demonstrado na Figura 33, a maioria dos avaliadores (67%) concordam totalmente em afirmar que o aplicativo segue um padrão de cores e estilos consistentes nas telas, seguidos de 33% que concordam parcialmente. Desta forma, infere-se que o padrão de cores das telas (preto) e botões (cinza), adotado em todas as telas do aplicativo, propiciou um resultado positivo no que diz respeito à consistência e padronização (H4).

Figura 33 - Resultado da pergunta 6



Fonte: elaborado pelo autor.

Os ícones foram utilizados somente no menu do aplicativo, a fim de facilitar a identificação da opção e evitar a escolha da opção errada. Neste sentido, perguntou-se aos

participantes se o aplicativo deixa claro através de seus ícones o objetivo da opção? com intuito de elucidar a eficácia do aplicativo perante a prevenção de erro (H5). A Figura 34 apresenta que 33% dos participantes concordam totalmente e 53% concordam parcialmente que o aplicativo deixa claro através de seus ícones o objetivo da opção, seguidos de 13% que discordam parcialmente da afirmação. Sendo assim, embora 33% dos participantes tenham concordado totalmente, observa-se um ponto fraco na prevenção de erros (H5).

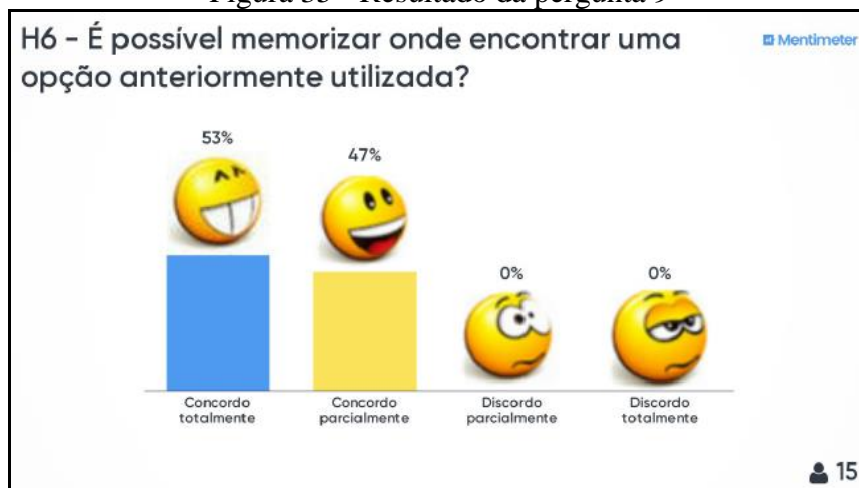
Figura 34 - Resultado da pergunta 7



Fonte: elaborado pelo autor.

Em relação a ajuda aos usuários para reconhecerem, diagnosticarem e se recuperarem de erros (H6), 53% dos participantes não encontraram dificuldades em se recordar das opções do aplicativo, seguidos de 47% que, possivelmente, tiveram alguma dificuldade de baixa relevância (Figura 35). O aplicativo possui poucas interfaces das quais o usuário necessita, e ainda, algumas que utiliza somente uma vez, como é o caso do cadastro do usuário. Sendo assim, observa-se que não há dificuldades na memorização das opções.

Figura 35 - Resultado da pergunta 9



Fonte: elaborado pelo autor.

Já com relação ao reconhecimento em vez de memorização (H7), 67% dos participantes entende que os objetos, ações e opções visíveis são compreensíveis no aplicativo e 33% concorda parcialmente com essa afirmação (Figura 36). Logo, com o resultado obtido pela pergunta o aplicativo possui teclas, imagens ou ícones para facilitar o seu uso conclui-se que o app permitiu o fácil reconhecimento da opção pelos avaliadores.

Figura 36 - Resultado da pergunta 8



Fonte: elaborado pelo autor.

Buscou-se identificar também a percepção dos participantes da pesquisa no que diz respeito a flexibilidade e eficiência de uso (H8) do aplicativo com a seguinte pergunta: As mensagens do aplicativo trazem informações relevantes ou na maioria das vezes necessárias? (Figura 37). Neste contexto, a Figura 37 traz um resultado positivo, no qual 67% dos avaliadores concordam totalmente e 33% concordam parcialmente que as mensagens trazem informações importantes. Assim sendo, observa-se que é essencial, durante a operação, que as mensagens sejam relevantes e claras para os usuários, respeitando um dos propósitos do aplicativo, poupar tempo.

Figura 37 - Resultado da pergunta 10



Fonte: elaborado pelo autor.

No que diz respeito ao design estético e minimalista (H9), foi perguntado aos participantes se o aplicativo possui mensagens de erros com linguagens simples (Figura 38). Neste sentido, observa-se um ponto de atenção, visto que 53% dos participantes compreendem parcialmente a linguagem utilizada nas mensagens do aplicativo, enquanto somente 33% concordam totalmente (Figura 38). Em contrapartida, segundo apresenta a Figura 38, os participantes que discordam parcialmente totalizam 13%, resultado do qual pode-se concluir que houverem algumas informações dispostas em diálogos ou mensagens, as quais não puderam ser compreendidas por parte dos participantes. De fato, algumas mensagens de erro inesperadas podem confundir o usuário quanto ao problema, como por exemplo uma falha de conexão com o servidor, porém, na maior parte dos tratamentos de exceção buscou-se, no mínimo, identificar a operação que resultou no erro, deixando evidenciado ao usuário qual a ação que falhou.

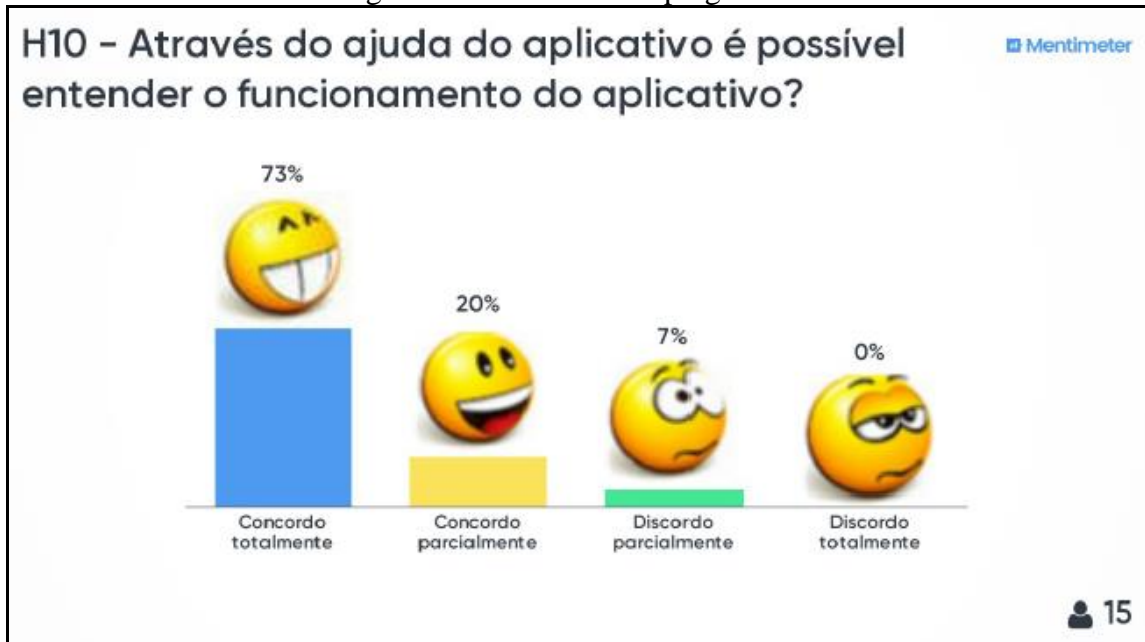
Figura 38 - Resultado da pergunta 11



Fonte: elaborado pelo autor.

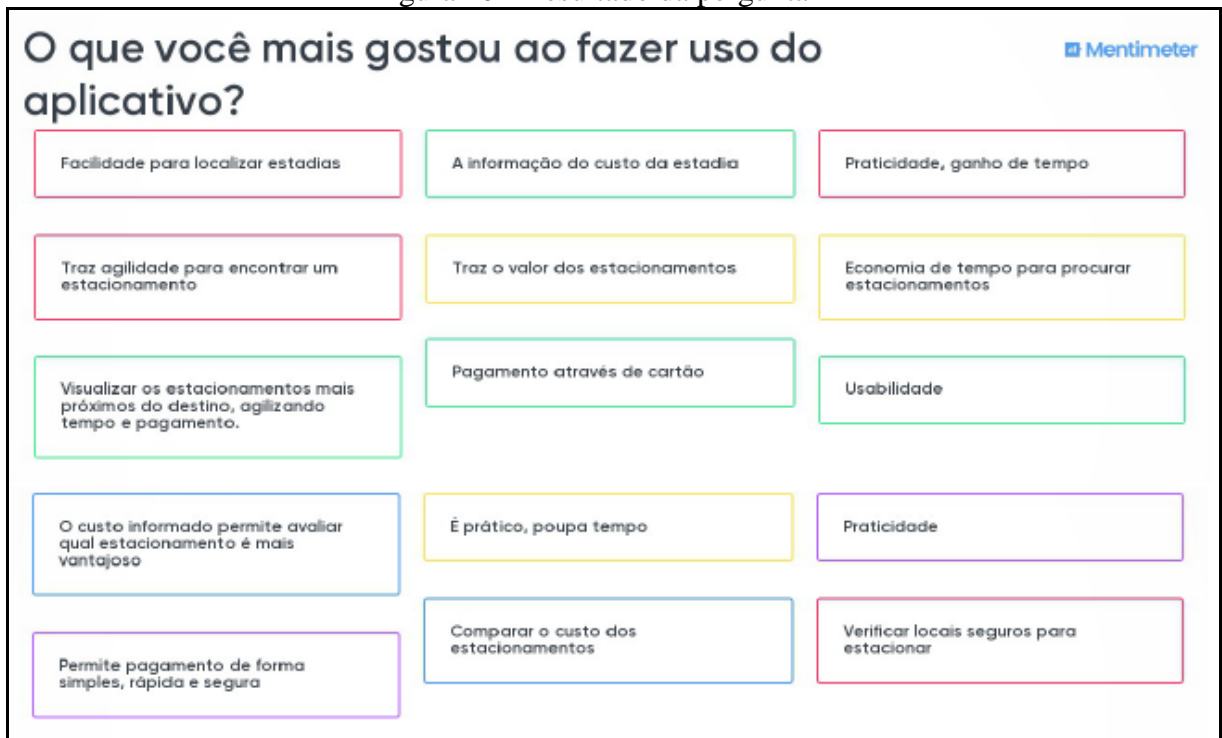
A utilidade da ajuda e documentação (H10) também foi apurada por meio da pergunta através da ajuda do aplicativo é possível entender o funcionamento do aplicativo? (Figura 39). A partir do resultado, é possível identificar que a opção ajuda pode ser melhorada, de forma que as dúvidas dos usuários sejam completamente sanadas. No total, 73% das respostas apontam a satisfação total perante o ajuda, seguido de 20% parcialmente satisfeitos. Em contrapartida, 7% informaram discordar parcialmente com relação ao entendimento do aplicativo por meio da ajuda do aplicativo, inferindo assim, que pudesse haver algum ponto de dúvida, cabendo a revisão das opções dispostas.

Figura 39 - Resultado da pergunta 12



Com o objetivo de obter mais informações acerca da percepção do usuário a respeito do aplicativo, conforme apresenta a Figura 40, lhes foi perguntado o que mais gostou ao fazer uso do aplicativo. Entre as respostas, pode-se notar que os avaliadores, em geral, ressaltam a facilidade de encontrar estacionamentos e comparar preços. Alguns usuários ainda citam que o aplicativo traz certa praticidade e lhes poupa tempo com suas funcionalidades.

Figura 40 - Resultado da pergunta 14



Por outro lado, também foi pedido aos avaliadores, conforme traz a Figura 41, que descrevessem o que menos gostaram de fazer ao usar o aplicativo. O resultado demonstra que grande parte dos avaliadores não identificou nada que lhes desmotivou no uso do aplicativo. Entretanto, observa-se um tópico recorrente indicando que há uma dificuldade para realizar a leitura do código de barras QR Code ao realizar uma estadia. Neste sentido, vale ressaltar que, dependendo do dispositivo móvel, pode haver uma melhora ou piora nesse aspecto, devido principalmente, as configurações de hardware do dispositivo. Ademais, pode-se inferir que alguns usuários, possivelmente, têm pouco prática com o uso desse tipo de tecnologia.

Figura 41 - Resultado da pergunta 15



Fonte: elaborado pelo autor.

Por fim, no final da pesquisa ainda foi avaliada a satisfação dos participantes em relação à experiência que tiveram, fazendo-lhes as seguintes perguntas a fim de identificar se utilizariam o aplicativo novamente e o recomendariam para amigos: Você usaria o aplicativo novamente? (Figura 42) e Você recomendaria o aplicativo para seus amigos? (Figura 43). O resultado apresentado na Figura 42 demonstra que 100% dos participantes voltariam a utilizar o aplicativo, concluindo-se que causou boa impressão no que diz respeito a sua utilidade e UX. Ademais, de acordo com a Figura 43, a maioria dos participantes (93%) recomendariam o aplicativo a um amigo, enquanto apenas 7% responderam que não o recomendariam. Desta forma, pode-se concluir que o resultado obtido com estas perguntas é satisfatório, pois comprova que o aplicativo cumpriu seus objetivos.



Figura 42 - Resultado da pergunta 16



Fonte: elaborado pelo autor.

Figura 43 - Resultado da pergunta 17



Fonte: elaborado pelo autor.

## 4 CONCLUSÕES

Este trabalho apresentou o desenvolvimento de um aplicativo para busca e pagamento de estacionamentos via dispositivo móvel, intitulado Estacione. A ideia do aplicativo emergiu a partir de uma dificuldade percebida no dia-dia do autor e de outras pessoas próximas, principalmente pelo fato de facilitar o pagamento de estacionamentos.

Os objetivos gerais apresentados na seção 1.1 foram alcançados. O aplicativo desenvolvido disponibilizou um mapa com informações dos estacionamentos credenciados, possibilitando os usuários buscarem por um lugar para estacionar, assim como permitiu que os usuários visualizassem o tempo de estadia no estacionamento e preço previsto por meio de um temporizador. Além disso, disponibilizou uma interface para credenciamento tanto do usuário quanto do estacionamento. Por fim, completando todos os objetivos propostos neste trabalho, permitiu que os usuários realizassem pagamento eletrônico pelo aplicativo.

A fundamentação teórica, apresentada na seção 2, auxiliou no entendimento inicial das principais tecnologias utilizadas. O estudo relacionado aos pagamentos móveis (seção 2.1) teve maior relevância pelo fato de contribuir na obtenção de uma visão mais ampla na aplicabilidade de sistemas com pagamento móvel, principalmente, devido aos números trazidos e às informações mundiais sobre o tema.

Apesar de ter-se demonstrado a existência de propostas semelhantes (seção 2.4), o trabalho desenvolvido justifica-se por mesclar a praticidade de encontrar estacionamentos e pagar suas estadias por meio de pagamento móvel, com a possibilidade de realizar isso com segurança por meio da leitura do QR Code. Cabe ainda salientar, que se observou uma gama de possibilidades nesse mercado, principalmente pelo fato de haver aplicativos consolidados e bem difundidos com as finalidades propostas neste trabalho. Além disso, esse trabalho contribui em apresentar o método Relationship of M3C with User Requirements and Usability and Communicability Assessment to groupware (M3C-URUCAg), que pode ser utilizado em trabalhos que queiram avaliar a usabilidade de aplicativos móveis com usuários finais sendo o especialista do aplicativo, bem como relacionar os requisitos com as heurísticas de Nielsen.

Ao longo do trabalho foram superadas dificuldades técnicas devido à pouca prática nas tecnologias empregadas, tais como: o Ionic, o Hibernate e o Heroku. Entretanto, as tecnologias escolhidas foram imprescindíveis para o êxito no resultado obtido.

### 4.1 EXTENSÕES

Como extensões deste trabalho, pode-se sugerir:

- a) dispor de uma opção para fazer a reserva de vagas;



- b) permitir que o usuário grave sua senha, sem que tenha que realizar login sempre que abre o aplicativo;
- c) calcular as rotas possíveis para chegar até o estacionamento desejado;
- d) dispor da distância do usuário perante um determinado estacionamento;
- e) integrar com cancelas para compor um sistema de estacionamento autônomo, ou seja, automático desde o controle de entrada até o pagamento no aplicativo;
- f) integração com ferramentas como Waze ou Google Maps para disponibilizar informações aos seus usuários sobre estacionamentos disponíveis;
- g) criar uma plataforma web para abranger mais tipos de usuários.

## REFERÊNCIAS

- ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO 9241-210**: Ergonomia da Interação humano-sistema parte 210: Projeto centrado no ser humano para sistemas interativos. 1 ed. Rio de Janeiro: ABNT, 2011. 3 p. Disponível em: <<https://www.target.com.br/produtos/normas-tecnicas/42352/nbriso9241-210-ergonomia-da-interacao-humano-sistema-parte-210-projeto-centrado-no-ser-humano-para-sistemas-interativos>>. Acesso em: 01 jul. 2018.
- ABRAHÃO, Ricardo de Sena. **Intenção de Adoção do Mobile Payment**: Uma Análise à Luz das Teorias de Aceitação e Uso de Tecnologia. 2015. 107 f. Dissertação (Mestrado) - Curso de Administração, Universidade Federal de Uberlândia, Uberlândia, 2015.
- ALHAK, Said Hikmat Abd. **Estacionamento Inteligente**. 2011. 76 f. TCC (Graduação) - Curso de Engenharia de Computação, Faculdade de Tecnologia e Ciências Sociais Aplicadas, Centro Universitário de Brasília, Brasília, 2011.
- BERNARDI, José Vicente Elias; LANDIM, Paulo M. Barbosa. **Aplicação do Sistema de Posicionamento Global (GPS) na coleta de dados**. DGA, IGCE, UNESP/Rio Claro, Lab. Geomatemática, Texto Didático, v. 10, n. 31, p. 2002, 2002.
- CAPGEMINI; PARIBAS, Bnp. **World Payments Report 2017**. [S.I.], 2017. Disponível em: <<https://www.worldpaymentsreport.com>>. Acesso em: 30 jan. 2018.
- COSTA, João Filipe Pires da. **Projecto de um Parque de Estacionamento**. 2008. 91 f. Dissertação (Mestrado) - Curso de Engenharia Civil, Departamento de Engenharia Civil, Faculdade de Engenharia da Universidade do Porto, Cidade do Porto, 2008.
- COSTA, Simone Erbs da. **Avaliação de Usabilidade e Comunicabilidade com a Experiência do Usuário pelo Método M3C-URUCAg**. iLibras Collaborative, 2018. Disponível em: <<https://www.ilibrascollaborative.com/post-unico/2018/06/20/Avalia%C3%A7%C3%A3o-de-Usabilidade-e-Comunicabilidade-pelo-m%C3%A9todo-M3C-URUCAg>>. Acesso em: 01 jul. 2018.
- CYBIS, Walter; BETIOL, Adriana Holtz; FAUST, Richard. **Ergonomia e usabilidade**: Conhecimentos, Métodos e Aplicações. 3a Edição ed., São Paulo, 2015.
- ECLIPSE FOUNDATION, **IDE**. [S.I.] 2018. Disponível em: <<https://www.eclipse.org/ide>>. Acesso em: 01 jun. 2018.
- ESTAPAR. **Vaga Inteligente**. [S.I.] 2018. Disponível em: <<https://www.estapar.com.br/aplicativo-vaga-inteligente>>. Acesso em: 29 jun. 2018.
- FERREIRA, Sara Cristina de Sousa. **As tecnologias móveis no auxílio ao estacionamento urbano**: o caso de Aveiro. 2014. 155 f. Dissertação (Mestrado) - Curso de Comunicação MultimédiaMultimídia, Departamento de Comunicação e Arte, Universidade de Aveiro, Aveiro, 2014.
- FOLHA DE SÃO PAULO. **Pagamento Eletrônico já está em 62% dos Shoppings de São Paulo**. São Paulo, 31 dez. 2013. Disponível em: <<http://www1.folha.uol.com.br/cotidiano/2013/12/1388733-pagamento-eletronico-ja-esta-em-62-dos-shoppings-de-sao-paulo.shtml>>. Acesso em: 30 jun. 2018.
- FOWLER, Martin. **UML Essencial: um breve guia para linguagem padrão**. Bookman Editora, 2014.

- GOOGLE DEVELOPERS, **Maps JavaScript API**. [S.I.] 2018. Disponível em: <<https://developers.google.com/maps/documentation/javascript/tutorial>>. Acesso em: 30 jun. 2018.
- HU, Wen-chenChen; LEE, Chung-wei; KOU, Weidong. **Advances in Security and Payments Methods for Mobile Commerce**. Londres: Idea Group Publishing, 2005. 351 p.
- INRIX INC. **ParkMe**. [S.I.], 5 jun. 2018. Disponível em: <<https://play.google.com/store/apps/details?id=com.parkme.consumer>>. Acesso em: 30 jun. 2018.
- INSTITUTO BRASILEIRO DE GEOGRAFIA ESTATÍSTICA – IBGE. **Frota municipal de veículos**. [S.I.], 2016. Disponível em: <<https://cidades.ibge.gov.br/brasil/pesquisa/22/28120>>. Acesso em: 30 jun. 2018.
- IONIC FRAMEWORK, **Core Concepts**. [S.I.] 2018. Disponível em: <<https://ionicframework.com/docs/intro/concepts>>. Acesso em: 30 jun. 2018.
- JET BRAINS, **Webstorm**. [S.I.], 2018. Disponível em: <<https://www.jetbrains.com/webstorm>>. Acesso em: 02 jun. 2018.
- MAI, Thuy. **Global Positioning System**. [S.I.], 2014. Disponível em: <<https://www.nasa.gov/directorates/heo/scan/communications/policy/GPS.html>>. Acesso em: 30 jun. 2018.
- MELLON, Byn. **Global Payments 2020: Transformation and Convergence**. [S.I.], 22 set. 2014. Disponível em: <<https://www.finextra.com/featurearticle/1966/global-payments-2020>>. Acesso em: 30 jun. 2018.
- MENTIMETER, **Plataforma**. [S.I.], 2018. Disponível em: <<https://www.mentimeter.com>>. Acesso em: 18 jun. 2018.
- MONICO, João Francisco Galera. **Posicionamento pelo NAVSTAR-GPS: Descrição, fundamentos e aplicações**. São Paulo: Unesp, 2000. 287 p.
- MOULIN, Robson. **O que é e como fazer uma Avaliação Heurística**. Design Interativo, 2013. Disponível em: <<http://www.designinterativo.etc.br/user-experience/o-que-e-e-como-fazer-uma-avaliacao-heuristica>>. Acesso em: 01 jul. 2018.
- NIELSEN, Jakob. Usability inspection methods. In: **Conference companion on Human factors in computing systems**. ACM, 1994. p. 413-414.
- PARKME INC. **Encontre & Pague pela Vaga Perfeita**. [S.I.], 2018. Disponível em: <<https://www.parkme.com/>>. Acesso em: 30 jun. 2018.
- PEREZ, Gilberto et al. The Use in Scale of Mobile Payment in Brazil: An Analysis of Trends Using the Delphi Method. In: INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND TECHNOLOGY MANAGEMENT – CONTECSI, 10., 2013, São Paulo. **Anais**. [São Paulo]: Universidade de São Paulo (USP), 2013. p. 1889 - 1910.
- PRATES, Raquel Oliveira; DINIZ, Simone Barbosa Junqueira. Capítulo 6 Avaliação de Interfaces de Usuário – Conceitos e Métodos. In: **Jornada de Atualização em Informática do Congresso da Sociedade Brasileira de Computação**, Rio de Janeiro: PUC-Rio, 2003. p. 1-49.
- ROEHRS, Alex. **4iPay: Modelo para Sistemas de Pagamento Móvel em Comércio Ubíquo**. 2012. 115 f. Dissertação (Mestrado) - Curso de Computação Aplicada, Universidade do Vale do Rio dos Sinos – Unisinos, São Leopoldo, 2012.

SALESFORCE, **Heroku**. [S.l.], 2018. Disponível em: <<https://www.heroku.com/platform>>. Acesso em: 30 maio 2018.

SOON, Tan Jin. QR Code. **The Synthesis Journal: Section three**. Singapura, p. 59-78. 2008.

SOURCEFORGE. DB Designer Fork. [S.l.], 2018. Disponível em: <<https://sourceforge.net/projects/dbdesigner-fork/>>. Acesso em: 24 jun. 2018.

## APÊNDICE A – Dicionário de Dados

Este Apêndice especifica as entidades apresentadas na seção 3.2.3. A seguir, elucida-se os tipos de dados utilizados.:

- a) `bigint`: tipo numérico que permite armazenar valores inteiros grandes;
- b) `integer`: tipo numérico que permite armazenar valores inteiros normais;
- c) `double`: tipo decimal que permite armazenar valores com casas decimais;
- d) `datetime`: armazena valores de datas e hora;
- e) `varchar`: armazena uma cadeia de caracteres de comprimento variável;
- f) `char`: armazena caractere de 1 byte.

O Quadro 18 traz a tabela `usuario` e os campos que a compõe.

Quadro 18 - Tabela `usuario`

Tabela: <code>usuario</code>		
Tabela responsável pelo armazenamento dos dados de usuários do aplicativo.		
Atributo	Tipo	Descrição
<code>id_usuario</code>	<code>Bigint</code>	Identificador único do usuário
<code>ds_login</code>	<code>Varchar (20)</code>	Login do usuário
<code>ds_email</code>	<code>Varchar (20)</code>	E-mail do usuário
<code>ds_senha</code>	<code>Varchar (20)</code>	Senha do usuário
<code>nm_usuario</code>	<code>Varchar (100)</code>	Nome do usuário
<code>fl_termos</code>	<code>Char (1)</code>	Termos e políticas aceitos
<code>tp_usuario</code>	<code>Char (1)</code>	Tipo do usuário

Fonte: elaborada pelo autor.

Já no Quadro 19 é apresentada a tabela para armazenamento dos `estacionamentos` credenciados.

Quadro 19 - Tabela `estacionamento`

Tabela: <code>estacionamento</code>		
Tabela responsável pelo armazenamento dos dados de estacionamentos credenciados ao aplicativo.		
Atributo	Tipo	Descrição
<code>id_estacionamento</code>	<code>Bigint</code>	Identificador único do estacionamento
<code>id_usuario</code>	<code>Bigint</code>	Chave Estrangeira (FK) do usuário
<code>nm_estacionamento</code>	<code>Varchar (200)</code>	Nome do estacionamento
<code>vl_longitude</code>	<code>Double</code>	Longitude da localização do estacionamento no mapa.
<code>ds_localizacao</code>	<code>Varchar (200)</code>	Latitude da localização do estacionamento no mapa
<code>ds_endereco</code>	<code>Varchar (200)</code>	Endereço do estacionamento
<code>vl_preco</code>	<code>Double (12,2)</code>	Preço cobrado no estacionamento

Fonte: elaborado pelo autor.

O Quadro 20 traz a tabela `estadia`, responsável pelo armazenando das `estadias` dos usuários.

Quadro 20 - Tabela estadia

Tabela: estadia		
Tabela responsável pelo armazenamento das estadias dos usuários nos estacionamentos.		
Atributo	Tipo	Descrição
id_estadia	Bigint	Identificador único da estadia
id_usuario	Bigint	Chave Estrangeira (FK) do usuário
id_estacionamento	Bigint	Chave Estrangeira (FK) do estacionamento
dt_entrada	Datetime	Data de entrada no estacionamento
dt_saida	Datetime	Data de saída do estacionamento
vl_preco	Double (12,2)	Valor pago na estadia no estacionamento
id_pagamento	Varchar (255)	Identificador único do pagamento

Fonte: elaborado pelo autor.

Por fim, no Quadro 21 é apresentada a tabela para armazenamento da forma de pagamento do usuário.

Quadro 21 - Tabela forma\_pagamento

Tabela: forma_pagamento		
Tabela responsável pelo armazenamento da forma de pagamento dos usuários.		
Atributo	Tipo	Descrição
id_formapagamento	Bigint	Identificador único da forma de pagamento
id_usuario	Bigint	Chave Estrangeira (FK) do usuário
nr_cartao	Integer (50)	Número do cartão de crédito
nm_cartao	Varchar (100)	Nome do proprietário do cartão de crédito
dt_validade	Varchar (10)	Data de validade do cartão de crédito
cd_seguranca	Varchar (3)	Código de segurança do cartão de crédito
ds_bandeira	Varchar (50)	Bandeira do cartão de crédito

Fonte: elaborado pelo autor.

## APÊNDICE B – Descrição dos casos de uso

Este apêndice traz a descrição dos principais casos de uso. Preliminarmente, o Quadro 22 apresenta a descrição do caso de uso UC01 - Criar usuário.

Quadro 22 - Descrição do caso de uso UC01 - Criar usuário

<b>Caso de uso</b>	<b>Criar usuário</b>
Descrição	Este caso de uso descreve os procedimentos realizados para criação de usuários.
Pré-condição	Aplicativo em execução e não estar logado com algum usuário.
Pós-Condição	USUARIO é criado e o login é realizado automaticamente no aplicativo, redirecionando-o para o mapa de busca.
Cenário Principal	O USUARIO pressiona o botão para criar uma conta. O aplicativo o redireciona para a tela de cadastro. O USUARIO preenche os campos necessários e pressiona o botão para cadastrar. O sistema cria a conta do usuário e o redireciona para o mapa principal do aplicativo.
Caminhos alternativos	O USUARIO não preenche todas as informações necessárias na tela e o sistema o aguarda preencher. O USUARIO cancela o cadastro por meio de botão para cancelar.

Fonte: elaborado pelo autor.

No Quadro 23 é apresentada a descrição do caso de uso UC02 - Realizar login.

Quadro 23 - Descrição do caso de uso UC02 - Realizar login

<b>Caso de uso</b>	<b>Realizar login</b>
Descrição	Este caso de uso descreve os procedimentos realizados para efetuar login no aplicativo.
Pré-condição	Aplicativo em execução e não estar logado com algum USUARIO.
Pós-Condição	USUARIO conectado no aplicativo e redirecionado para o mapa de busca.
Cenário Principal	O USUARIO preenche seu login e senha e pressiona o botão para entrar. O aplicativo o redireciona para a tela de cadastro. O USUARIO preenche os campos necessários e pressiona o botão para cadastrar. O sistema cria a conta do USUARIO e o redireciona para o mapa principal do aplicativo.
Cenário de exceção	Usuário ou senha incorreto. Volta para o passo 1 do cenário principal.

Fonte: elaborado pelo autor.

Já no Quadro 24 é apresentado o caso de uso UC04 - Manter forma de pagamento.

Quadro 24 - Descrição do caso de uso UC04 - Manter forma de pagamento

<b>Caso de uso</b>	<b>Manter forma de pagamento</b>
Descrição	Este caso de uso descreve os procedimentos realizados para o USUARIO cadastrar uma forma de pagamento.
Pré-condição	Usuário cadastrado e conectado no aplicativo.
Pós-Condição	Forma de pagamento cadastrada no aplicativo.
Cenário Principal	O USUARIO preenche os dados do seu cartão e pressiona o botão para gravar. O sistema grava os dados.
Cenário alternativo	O USUARIO não preenche todos os dados e o sistema aguarda o preenchimento. O USUARIO pressiona o botão para cancelar.

Fonte: elaborado pelo autor.

Ainda, no Quadro 25 é trazida a descrição do caso de uso UC05 - Manter perfil.

Quadro 25 - Descrição do caso de uso UC05 - Manter perfil

<b>Caso de uso</b>	<b>Manter perfil</b>
Descrição	Este caso de uso descreve os procedimentos realizados para o USUARIO manter seu cadastro.
Pré-condição	USUARIO cadastrado e conectado no aplicativo.
Pós-Condição	USUARIO alterado.
Cenário Principal	O USUARIO altera os dados permitidos na sua conta. O sistema grava os dados.
Cenário alternativo	O USUARIO pressiona o botão para cancelar.

Fonte: elaborado pelo autor.

No

Quadro 26 apresenta-se a descrição do principal caso de uso UC06 - Registrar entrada e saída.

Quadro 26 - Descrição do caso de uso UC06 - Registrar entrada e saída

<b>Caso de uso</b>	<b>Registrar entrada e saída</b>
Descrição	Este caso de uso descreve os procedimentos realizados para o USUARIO registrar sua entrada e saída do estacionamento.
Pré-condição	USUARIO cadastrado e conectado no aplicativo.
Pós-Condição	Uma estadia iniciada ou finalizada.
Cenário Principal	O USUARIO pressiona o botão para iniciar uma estadia. O sistema abre a câmera do dispositivo móvel para ler o QR Code. O USUARIO realiza a leitura do QR Code. O sistema identifica o estacionamento e inicia uma estadia. O sistema apresenta uma mensagem avisando que o checkin foi realizado. O sistema apresenta um temporizador com o tempo decorrido e o preço estimado da estadia.
Cenário 2 (estadia em andamento)	O USUARIO pressiona o botão para terminar a estadia previamente iniciada. O sistema abre a câmera do dispositivo móvel para ler o QR Code. O USUARIO realiza a leitura do QR Code. O sistema termina a estadia do USUARIO no estacionamento. O sistema apresenta uma mensagem avisando que o checkout foi realizado.

Fonte: elaborado pelo autor.



**APÊNDICE C – Tarefas da avaliação**

Este apêndice traz uma sequência de tarefas que foram elaboradas para o usuário realizar os testes do aplicativo desenvolvido. No Quadro 27 é apresentada a relação de procedimentos necessários para realizar posteriormente a avaliação do aplicativo desenvolvido.

Quadro 27 - Tarefas da Avaliação de Usabilidade e UX pelo Método M3C-URUCAg

Tarefas preliminares pré-avaliação
<p><b>Objetivos da aplicação:</b> As tarefas elencadas em seguida têm como objetivo dar subsidio ao avaliador, de forma com que obtenha uma opinião sobre o aplicativo desenvolvido e, após realizadas, possa responder o questionário de avaliação.</p>
<p><b>Tarefas que deverão ser realizadas:</b></p> <p>1 - Faça login na aplicação com os dados abaixo: login = estacione senha = estacione2018 Visualizar informações dos estacionamentos:</p> <p>2 - Localize sua posição atual no mapa. 2.1 - Expanda e contraia o mapa, a fim de encontrar algum estacionamento (marcador no mapa). 2.1.1 - Encontrando um estacionamento, e pressione o marcador. Pesquisar localidades no mapa, a fim de encontrar estacionamentos:</p> <p>3 - Na barra superior do mapa, digite uma localização para pesquisar. 3.1 - Selecione entre as opções dispostas na lista, a qual se refere à sua pesquisa.</p> <p>Iniciar uma estadia no estacionamento: 4 - Arraste o dedo da esquerda para direito, ou, pressione o botão no canto esquerdo superior para abrir o menu do aplicativo. 4.1 - Navegue até a opção "Nova Estadia" e pressione-a. 4.2 - Pressione o botão "Iniciar". 4.2.1 - Posicione o smartphone para leitura do código de barras (QRCode). 4.3 - Aguarde 1 minuto. 4.4 - Pressione o botão "Terminar". 4.4.1 - Posicione o smartphone para leitura do código de barras (QRCode).</p> <p>Verificar histórico de estadias: 5 - Arraste o dedo da esquerda para direito, ou, pressione o botão no canto esquerdo superior para abrir o menu do aplicativo. 5.1 - Navegue até a opção "Histórico" e pressione-a. 5.2 - Identifique as informações da estadia realizada anteriormente.</p> <p>Inserindo um cartão de crédito para pagamento: 6 - Arraste o dedo da esquerda para direito, ou, pressione o botão no canto esquerdo superior para abrir o menu do aplicativo. 6.1 - Navegue até a opção "Pagamento" e pressione-a. 6.2 - Insira os dados abaixo nos campos: Número do Cartão: 4455668898563211 Nome do Cartão: Cartão Teste Estacione Validade: 05 2019 Código de Segurança: 123 6.2.1 - Pressione o botão "Gravar".</p> <p>Sair do aplicativo: 7 - Arraste o dedo da esquerda para direito, ou, pressione o botão no canto esquerdo superior para abrir o menu do aplicativo.</p>

7.1 - Navegue até a opção "Sair" e pressione-a.

Criar usuário:

8 - Na tela de login, localize o botão "Não tem conta? Cadastre-se" e pressione-o.

8.2 - Arraste o dedo na tela da direita para a esquerda.

8.3 - Arraste o dedo na tela da direita para a esquerda mais uma vez.

8.4 - Localize o botão "Para Motoristas" e pressione-o.

8.5.1 - Digite no campo identificado como "Digite seu e-mail" o e-mail usuario1@gmail.com.

8.5.2 - Digite no campo identificado como "Digite seu nome" o nome "Usuário Teste" (sem aspas).

8.5.3 - Digite no campo identificado como "Digite um login" o login "usuario1" (sem aspas).

8.5.4 - Digite no campo identificado como "Digite uma senha" a senha "123" (sem aspas).

8.5.5 - Digite no campo identificado como "Confirme sua senha" a senha "123" (sem aspas).

8.5.6 - Marque a opção "Aceito os Termos e Políticas".

8.5.7 - Pressione o botão "Cadastrar".

Explorar as opções disponíveis na aplicação:

9 - Navegue no mapa, expandindo-o e contraindo-o.

10 - Navegue nos menus do aplicativo.

Fonte: elaborado pelo autor.

## APÊNDICE D – Termo de Consentimento Livre e Esclarecido (TCLE)

Este apêndice apresenta o TCLE disponibilizado para os participantes da avaliação antes dela ocorrer (Figura 44).

Figura 44 - Termo de Consentimento Livre e Esclarecido

### Termo de Consentimento Livre e Esclarecido

---

O(a) senhor(a) está sendo convidado a participar de uma pesquisa de graduação, intitulada "Avaliação de usabilidade do aplicativo Estacione" que fará uma breve avaliação com o objetivo geral de encontrar e pagar estacionamentos com cartão de crédito. Esta avaliação será realizada no site MentiMeter.

O professor regente fará a explicação da forma que será aplicado a avaliação. Como esta é uma participação voluntária, o(a) Senhor(a) e seu/sua acompanhante não terão despesas e nem serão remunerados pela participação na pesquisa. Todas as despesas decorrentes de sua participação serão ressarcidas. Em caso de danos, decorrentes da pesquisa será garantida a indenização. Os possíveis desconfortos e riscos decorrentes do estudo, levando-se em conta que é uma pesquisa, e os resultados positivos ou negativos somente serão obtidos após a sua realização. Assim, estou sujeito a realização de tarefas pré-definidas e especificadas no formulário de avaliação. Além disso, a minha avaliação poderá ou não ser considerada no resultado final do aplicativo, dependendo da forma que eu estarei respondendo minha avaliação.

Estou ciente que minha privacidade será respeitada, ou seja, meu nome ou qualquer outro dado ou elemento que possa, de qualquer forma, me identificar, será mantido em sigilo. Também fui informado que eu posso me recusar a participar do estudo, ou retirar meu consentimento a qualquer momento, sem precisar justificar, e que, por desejar sair da pesquisa, não sofrerei qualquer prejuízo.

Os pesquisadores envolvidos no estudo são, Paulo Arnoldo Koglin Junior, que posso entrar em contato pelo e-mail pk.koglin@yahoo.com.br e da orientadora da pesquisa Simone Erbs da Costa, que posso entrar em contato pelo e-mail simoneerbsdacosta@gmail.com. Além disso, é assegurada toda assistência durante toda a pesquisa, bem como me é garantido o livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências, ou seja, tudo que eu queria saber antes, durante e depois da minha participação.

Dessa forma, tendo sido orientado quanto ao teor de todo aqui mencionado e compreendido a natureza e o objetivo do referido estudo, manifesto meu livre consentimento em participar, estando totalmente ciente de que não existe nenhum valor econômico, a receber ou a pagar, por minha participação. Caso exista algum dano decorrente a minha participação no estudo, serei devidamente indenizado conforme determina a lei.

Em caso de reclamação ou qualquer outra denúncia sobre esse estudo devo entrar em contato com a orientadora da pesquisa Simone Erbs da Costa pelo e-mail simoneerbsdacosta@gmail.com. Os benefícios e vantagens em participar deste estudo estão relacionados a contribuir para o uso de novas metodologias em salas de aula, tornando o ambiente de ensino aprendizagem mais próximo da realidade que estou inserido. A pessoa que estará acompanhando os procedimentos será o professor regente da disciplina que o questionário está sendo aplicado. O(a) senhor(a) poderá se retirar do estudo a qualquer momento, sem qualquer tipo de constrangimento. Solicitamos a sua autorização para o uso de seus dados para a produção de artigos técnicos e científicos.

A sua privacidade será mantida por meio da não-identificação do seu nome. Caso esteja de acordo, com o termo basta seguir o roteiro da pesquisa abaixo e responder as perguntas do questionário de avaliação, acessando o <https://www.menti.com/c8f2e684>.

Fonte: elaborado pelo autor.

## APÊNDICE E – Tarefas da avaliação

Este apêndice traz os emoticons utilizados na avaliação de usabilidade e experiência de usuários conforme o Método M3C-URUCAg (Figura 45).

Figura 45 - Emoticons aplicados na escala Likert



Fonte: Costa (2018).