

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ – UTFPR
CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE
SISTEMAS

TIAGO WOTTRICH

**DESENVOLVIMENTO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS
BASEADOS EM PLATAFORMAS CRUZADAS**

TRABALHO DE DIPLOMAÇÃO

MEDIANEIRA

2016

TIAGO WOTTRICH

**DESENVOLVIMENTO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS
BASEADOS EM PLATAFORMAS CRUZADAS**

Trabalho de Diplomação apresentado à disciplina de Trabalho de Diplomação, do Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas – COADS – da Universidade Tecnológica Federal do Paraná – UTFPR, como requisito parcial para obtenção do título de Tecnólogo.

Orientador: Prof. Ricardo Sobjak.

MEDIANEIRA

2016



Ministério da Educação
Universidade Tecnológica Federal do Paraná
Diretoria de Graduação e Educação Profissional
Curso Superior de Tecnologia em Análise e
Desenvolvimento de Sistemas



TERMO DE APROVAÇÃO

DESENVOLVIMENTO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS BASEADOS EM PLATAFORMAS CRUZADAS

Por
TIAGO WOTTRICH

Este Trabalho de Diplomação (TD) foi apresentado às 11:00h do dia 09 de junho de 2016 como requisito parcial para a obtenção do título de Tecnólogo no Curso Superior de Tecnologia em Manutenção Industrial, da Universidade Tecnológica Federal do Paraná, *Campus* Medianeira. O acadêmico foi arguido pela Banca Examinadora composta pelos professores abaixo assinados. Após deliberação, a Banca Examinadora considerou o trabalho aprovado com louvor e mérito.

Prof. Ricardo Sobjak
UTFPR – *Campus* Medianeira
(Orientador)

Prof.
UTFPR – *Campus* Medianeira
(Convidado)

Prof.
UTFPR – *Campus* Medianeira
(Convidado)

Prof.
UTFPR – *Campus* Medianeira
(Responsável pelas atividades de TCC)

RESUMO

A folha de aprovação assinada encontra-se na Coordenação do Curso.

WOTTRICH, Tiago. DESENVOLVIMENTO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS BASEADOS EM PLATAFORMAS CRUZADAS. TRABALHO DE DIPLOMAÇÃO - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Medianeira, 2016.

Mais que nunca “tempo é dinheiro”, e por essa ótica o conceito de desenvolvimento de plataformas cruzadas para aplicações móveis se encaixa bem. Não mais é necessário reescrever o mesmo código em diferentes linguagens para cada plataforma que deseja atingir, obtendo-se assim produtividade e economizando dinheiro. Este projeto tem como objetivo investigar este tipo de desenvolvimento juntamente com as principais ferramentas disponíveis no mercado, além de construir um protótipo de aplicação utilizando uma das tecnologias estudadas.

Palavras-chave: Desenvolvimento *Mobile*, *Xamarin*, *Apache Cordova*, *Delphi 10*, *Unity*.

ABSTRACT

WOTTRICH, Tiago. DESENVOLVIMENTO DE APLICAÇÕES PARA DISPOSITIVOS MÓVEIS BASEADOS EM PLATAFORMAS CRUZADAS. TRABALHO DE DIPLOMAÇÃO - Curso de Tecnologia em Análise e Desenvolvimento de Sistemas, Universidade Tecnológica Federal do Paraná. Medianeira, 2016.

More than ever “time is money”, and because of this point of view the concept of alternative mobile development fits well. It’s not necessary anymore having to rewrite the same code in different languages for each platform you wish to achieve, and because of that you gain productivity and saves money. This project has the objective of contextualize this kind of development and also the main tools available in the market. Lastly a prototype will be built making use of one of the technologies studied.

Keywords: Mobile development, Xamarin, Apache Cordova, Delphi 10, Unity.

LISTA DE FIGURA

Figura 1 – Diagrama de caso de uso.	26
Figura 2 – Diagrama de sequencia (cadastro viagem).....	27
Figura 3 – Diagrama de sequencia (finalizar viagem).	27
Figura 4 – Diagrama de atividades.....	28
Figura 5 – Diagrama de classes.....	29
Figura 6 – MER do protótipo.	30
Figura 7 – Diagrama de telas.	31
Figura 8 – Criação projeto Delphi 10.....	32
Figura 9 – Opções de aplicação Delphi 10.....	33
Figura 10 – Interface visual de componentes Delphi 10.....	34
Figura 11 – Código de alerta Delphi 10.....	34
Figura 12 – Executar aplicação Delphi 10.....	35
Figura 13 – Tela da aplicação Delphi 10.	35
Figura 14 – Mensagem da aplicação Delphi 10.	36
Figura 15 – Criação projeto Apache Cordova.	37
Figura 16 – Tipo de projeto Apache Cordova.....	37
Figura 17 – Estrutura de projeto Apache Cordova.	38
Figura 18 – Exemplo código aplicação Apache Cordova.	38
Figura 19 – Executar aplicação Apache Cordova.	39
Figura 20 – Tela aplicação Apache Cordova.....	39
Figura 21 – Mensagem da aplicação Apache Cordova.....	40
Figura 22 – Criação projeto Unity.....	41
Figura 23 – Tipo do projeto Unity.	41
Figura 24 – Novo botão Unity.....	42
Figura 25 – Novo texto Unity.....	42
Figura 26 – Novo script Unity.	43
Figura 27 – Código botão Unity.....	43
Figura 28 – Atribuição de evento para botão Unity.	44
Figura 29 – Selecionar plataforma Unity.	45
Figura 30 – Tela aplicação Unity.	46
Figura 31 – Tela mensagem Unity.	46
Figura 32 – Criação projeto Xamarin.....	47
Figura 33 – Tipo de projeto Xamarin.	47
Figura 34 – Estrutura projeto Xamarin.	48
Figura 35 – Adicionar arquivo XAML.....	48
Figura 36 - Selecionar arquivo XAML.....	49
Figura 37 – Exemplo código botão Xamarin.	49
Figura 38 – Exemplo código alerta Xamarin.	49
Figura 39 – Tela aplicação Xamarin.....	50
Figura 40 – Tela alerta aplicação Xamarin.....	50

Figura 41 – Tela de login.....	51
Figura 42 – Mensagem 1 login.....	52
Figura 43 – Mensagem 2 login.....	53
Figura 44 – Tela inicial.....	54
Figura 45 – Tela de configurações.....	55
Figura 46 – Tela de pessoas.....	56
Figura 47 – Consulta de pessoas.....	57
Figura 48 – Cadastro de pessoa.....	58
Figura 49 – Cadastro de pessoa parte 2.....	58
Figura 50 – Validação cadastro de pessoa.....	60
Figura 51 - Validação cadastro de pessoa parte 2.....	60
Figura 52 – Mensagem CPF inválido.....	61
Figura 53 – Mensagem e-mail inválido.....	61
Figura 54 – Mensagem e-mail já cadastrado.....	62
Figura 55 – Tela de veículos.....	63
Figura 56 – Consulta de veículos.....	64
Figura 57 – Cadastro de veículo.....	64
Figura 58 – Validação cadastro de veículo.....	66
Figura 59 – Mensagem placa já registrada.....	66
Figura 60 – Tela de viagens.....	67
Figura 61 – Tela de filtros de viagem.....	68
Figura 62 – Tela cadastro de viagem.....	69
Figura 63 – Tela cadastro de viagem parte 2.....	70
Figura 64 – Validação de viagem.....	71
Figura 65 – Validação de viagem parte 2.....	71
Figura 66 – Mensagem veículo não informado.....	72
Figura 67 – Mensagem pessoa não incluída.....	72
Figura 68 – Mensagem pessoa não é motorista.....	73
Figura 69 – Mensagem limite de pessoas excedido.....	73
Figura 70 – Mensagem GPS inativo.....	74
Figura 71 – Tela finalização viagem.....	75
Figura 72 - Tela finalização viagem parte 2.....	75
Figura 73 – Validação finalização viagem.....	76
Figura 74 - Validação finalização viagem parte 2.....	77
Figura 75 – Mensagem data final menor que incial.....	77
Figura 76 – Mensagem quilometragem final menor que incial.....	78
Figura 77 – Tela visualização viagem.....	79
Figura 78 - Tela visualização viagem parte 2.....	79
Figura 79 - Tela visualização viagem parte 3.....	80
Figura 80 – Tela mapa viagem.....	80

LISTA DE SIGLAS

AMPS	<i>Advanced Mobile Phone System</i>
APIs	<i>Application Programming Interface</i>
CPF	Cadastro de pessoas físicas
CSS	<i>Cascading Style Sheets</i>
GPS	<i>Global Positioning System</i>
HTML	<i>HyperText Markup Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IBM	<i>International Business Machines</i>
IDE	<i>Integrated Development Environment</i>
JSON	<i>JavaScript Object Notation</i>
PDA	<i>Personal Digital Assistant</i>
RA	Registro Acadêmico
SMS	<i>Short Message Service</i>
USB	<i>Universal Serial Bus</i>
UTFPR	Universidade Tecnológica Federal do Paraná
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVO GERAL	11
1.2	OBJETIVOS ESPECÍFICOS	11
1.3	JUSTIFICATIVA	12
2	REVISÃO BIBLIOGRÁFICA	13
2.1	CELULARES	13
2.2	SMARTPHONES	14
2.3	SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS	15
2.3.1	Windows Phone	15
2.3.2	iOS	15
2.3.3	Android	16
2.4	LOJAS VIRTUAIS	16
2.5	TIPOS DE DESENVOLVIMENTO	17
2.6	PLATAFORMAS CRUZADAS PARA DESENVOLVIMENTO DE APLICAÇÕES	18
2.6.1	Xamarin	18
2.6.2	Apache Cordova	19
2.6.3	Unity	20
2.6.4	Delphi 10	20
3	MATERIAL E MÉTODOS	22
3.1	FERRAMENTAS UTILIZADAS	22
3.2	ESTUDO DAS PLATAFORMAS	24
3.3	ESTUDO DO PROTÓTIPO	25
3.3.1	Diagrama De Caso de Uso	25
3.3.1	Diagramas De Sequência	26
3.3.2	Diagrama de Atividade	28
3.3.3	Diagrama de Classe	28
3.3.4	Modelo De Dados– MER	30
3.3.5	Diagrama de Telas	31
4	RESULTADOS E DISCUSSÃO	32
4.1	INVESTIGAÇÃO DAS PLATAFORMAS	32

4.1.1	Projeto Demonstrativo Em Delphi 10.....	32
4.1.2	Projeto Demonstrativo Em Apache Cordova	36
4.1.3	Projeto Demonstrativo Em Unity.....	40
4.1.4	Projeto Demonstrativo Em Xamarin.....	47
4.1.5	Descrição De <i>Layout</i> Das Telas	51
5	CONSIDERAÇÕES FINAIS	81
5.1	CONCLUSÕES.....	81
5.2	TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO.....	83
6	REFERÊNCIAS BIBLIOGRÁFICAS	84
7	APÊNDICE A – CASOS DE USO	89

1 INTRODUÇÃO

Existe uma maneira de acelerar o tempo de produção, baratear o custo e escrever o código de uma aplicação *mobile* apenas uma vez. Com o advento de ferramentas e tecnologias para desenvolvimento de plataformas cruzadas, isso é possível.

Desenvolvimento nativo para celulares é aquele onde se é voltado para uma plataforma específica, como *Android*, por exemplo. Por isso se é possível interagir e tirar vantagem do *hardware*, sistema operacional utilizado e outros programas que são típicos da plataforma. Sendo assim, para um aplicativo desenvolvido nativamente existir em duas plataformas diferentes é necessário que o mesmo seja escrito duas vezes em suas respectivas linguagens, o que torna longo o tempo de produção e o custo de desenvolvimento (APPEL, 2014).

Existem, porém, meios alternativos de desenvolvimento de aplicações para celulares. Neles é possível escrever o código de uma aplicação uma única vez e utiliza-lo para diversas plataformas, ou ao menos reescrever o mesmo código para cada plataforma que for utilizar, porém mantendo a mesma linguagem de programação.

Dentro da linha de meios alternativos de desenvolvimento que são os de plataforma cruzada, existem os aplicativos híbridos, que, assim como *sites da internet*, podem se utilizar de tecnologias *web* como HTML, CSS e *Java Script*. A primordial diferença é a capacidade de utilizarem de uma aplicação nativa que faz uso do navegador da plataforma escolhida, seja ela *Android*, *iOS* ou *Windows Phone*. Imagina-se isso como uma aplicação que funciona dentro do navegador nativo da plataforma, mas em uma versão de tela cheia, permitindo assim acesso a funções nativas como o acelerômetro, contatos e mais (KORF, OKSMAN, 2015).

Aplicativos híbridos são como qualquer outro aplicativo que você encontrará em seu celular. Com eles é possível fazer quase tudo que um aplicativo nativo faz, como: jogar, encontrar amigos em mídias sociais, tirar fotos, ler impressões digitais, lidar com notificações, GPS (*Global Positioning System*), mídias como áudios e vídeos e outras funcionalidades (BRISTOWE, 2015).

Uma aplicação híbrida bem escrita não deve parecer ou se comportar diferente de nenhuma maneira ou aspecto que seu equivalente nativo, e a tendência é que cada vez mais isso se torne verdade (BRISTOWE, 2015).

Este projeto tem como objetivo investigar e caracterizar os principais meios de desenvolvimento de plataformas cruzadas para aplicações de celulares, descrevendo as principais tecnologias e ferramentas do mercado atual, e por fim a confecção de um protótipo de aplicativo que realiza o controle de viagens para servidores da instituição UTFPR campus Medianeira.

1.1 OBJETIVO GERAL

Investigar as principais características do desenvolvimento de aplicativos em plataformas cruzadas, juntamente com os principais meios e ferramentas disponíveis, para então confeccionar um aplicativo utilizando uma das tecnologias existentes aplicando os conhecimentos obtidos.

1.2 OBJETIVOS ESPECÍFICOS

Os principais objetivos específicos são:

- Realizar uma revisão bibliográfica sobre desenvolvimento alternativo para celulares;
- Investigar os principais meios existentes deste tipo de desenvolvimento, como: Xamarin, Apache Cordova, Delphi 10 e Unity;
- Investigar características, pontos positivos e negativos utilizando uma das plataformas abordadas no tópico anterior;
- Confeccionar um aplicativo de protótipo fazendo uso de uma das plataformas estudadas dos conhecimentos obtidos para a UTFPR Medianeira.

1.3 JUSTIFICATIVA

Vive-se hoje em uma era interconectada, onde surgiu há nova sociedade, a sociedade em rede (CASTELLS, 2005). Essa sociedade não se limita a algum país ou região, e sim ao mundo todo, basta estar conectado à internet. Neste novo sistema social reina juntamente com o sentimento de individualidade, reina o de coletividade (PÁSCOA, 2011).

O advento da tecnologia de maneira geral ajudou ainda mais o desenvolvimento dessa sociedade coletiva. Dentro desse contexto de cultura de solidariedade e compartilhamento, o conceito de aplicações de plataformas cruzadas se encaixa muito bem. Há ferramentas hoje para este tipo de desenvolvimento que mesmo um leigo no assunto consegue utilizar, dar vida e compartilhar sua ideia.

Vive-se também em uma sociedade em que mais que nunca se fez valer a frase “Tempo é dinheiro”. Benjamin Franklin foi um intelectual do século XVIII, e uma de suas frases marcantes que ultrapassam fronteiras é: “*Remember that time is Money*”, ou seja, “lembrem-se que tempo é dinheiro”. Tempo é o bem mais precioso que o ser humano pode possuir, inclusive para aqueles que pensam ter nada (FILHO, 2012). Tempo impacta diretamente na produtividade, então é natural procurar sempre novas formas de aumentar a produtividade fazendo-se uso do mesmo período de tempo ou até diminuindo-o.

O tema de desenvolvimento móvel para plataformas cruzadas foi escolhido justamente por atender este requisito tão importante da sociedade. Esta maneira de desenvolvimento não é tão recente e existem diversas tecnologias e ferramentas disponíveis sobre este tema, depende-se então do perfil do programador em se identificar com alguma delas.

Para a parte prática do trabalho será feito um estudo sobre o processo de automação para controlar viagens feitas por servidores da UTFPR Medianeira e aplicar os conhecimentos obtidos para a confecção de uma aplicação fazendo uso de uma das tecnologias disponíveis para tal, levando-se em conta qual mais se adequa aos requisitos do protótipo em questão.

2 REVISÃO BIBLIOGRÁFICA

Esse capítulo tem como objetivo apresentar os conceitos sobre as tecnologias e metodologias a serem utilizadas no trabalho.

2.1 CELULARES

Até o ano de 1973 a telefonia móvel era limitada a telefones instalados em carros. Martin Cooper, um pesquisador e executivo da Motorola, em três de abril de 1973, fez história realizando a primeira ligação de um telefone móvel. O protótipo pesava 1.1kg medindo 23cm de altura, suportando 30 minutos de conversa e necessitando 10 horas para carregá-lo novamente (ANJARWALLA, 2009).

O primeiro sinal analógico de celular amplamente difundido foi nos Estados Unidos no ano de 1983. A *Advanced Mobile Phone System* (AMPS), foi a tecnologia pioneira que ajudou a expandir para o mercado a utilização de celulares, porém possuía severas falhas de segurança e a susceptibilidade de clonagem (AT&T, N/I).

Já no começo dos anos 90 foi introduzida a tecnologia de sinal digital, propiciando qualidade e segurança, além do revolucionário SMS (*Short Message Service*). Existiam dois sistemas que competiam entre si, um europeu e um americano. Com o advento do sinal digital surgiu também os serviços de telefonia pré-paga, que tornou o SMS a escolha de comunicação da juventude da época.

Até 2002, dificilmente era possível a instalação de outros aplicativos que já não vinham de fábrica do próprio celular. Isso mudou com a chegada dos primeiros celulares com suporte a Java (J2ME). Eles possuíam capacidade de fazer download de aplicativos e instala-los no celular, o que abriu uma oportunidade enorme não apenas para os fabricantes, mas também outras empresas ou qualquer programador que quisessem se aventurar no mundo dos celulares.

2.2 SMARTPHONES

Em 1993 IBM *Simon Personal Communicator*, ou simplesmente IBM Simon foi introduzido. Sendo este provavelmente o primeiro smartphone do mundo. Além de enviar e receber chamadas telefônicas, ele podia receber fax, e-mails, funcionar como um *pager* (Bipe, em português) e um PDA (*Personal Digital Assistant*). Possuía também aplicativos como: agenda de endereços, calendário, agenda de compromissos, calculadora, relógio mundial e anotações por meio de seu *display touchscreen* (SAGER, 2012). Seu sistema de arquivos era o *Datalight ROM-DOS*, introduzido em 1989 como um sistema operacional *MS-DOS* criado para sistemas embarcados (O'MALEY, 1994). Chegou a ser vendido cerca de 50 mil unidades, seu custo na época era de 899 dólares e sua bateria durava aproximadamente uma hora (BBC, 2014).

Um *smartphone* é um telefone móvel com um sistema operacional avançado que une características de um computador pessoal com características de tecnologias móveis ou uso manual. Possui grande capacidade de processamento e vasta variedade de aplicativos disponíveis para consumo. Tipicamente combina um celular comum com vários outros dispositivos móveis, como: tocador de música/vídeo, GPS (*Global Positioning System*), câmera digital e acesso à *internet*.

Basicamente a maneira de distinguir *smartphones* de telefones celulares comuns é simples. *Smartphones* possuem sistemas operacionais complexos, já um celular comum, não. Mesmo que nos dias atuais grande parte dos modelos com *touchscreen* seja realmente de *smartphones*, é possível encontrar celulares normais com este tipo de tela, por exemplo.

No terceiro trimestre de 2012, um bilhão de *smartphones* estavam em uso no mundo inteiro (REISINGER, 2012), e pela primeira vez, em 2013 foram vendidos mais *smartphones* que celulares comuns (NEWSHUB, 2013).

2.3 SISTEMAS OPERACIONAIS PARA DISPOSITIVOS MÓVEIS

O sistema operacional *Symbian* era o mais popular em uma escala global até o fim de 2010. Lançado em 2002, estava na maioria dos telefones Nokia e difundido amplamente na Europa e Ásia. Foi um celular com *Symbian* o primeiro a ser realmente feito propaganda como um “*smartphone*”.

Globalmente existiram inúmeros sistemas operacionais para smartphones que fizeram sucesso, como foi o caso do *Symbian*, *Windows Mobile*, *webOS*, entre outros. Porém apenas três detém 99,5% do mercado mundial, segundo dados do ultimo trimestre de 2015 (GARTNER, 2016), sendo estes: *Android*, *iOS* e *Windows Phone*.

2.3.1 Windows Phone

Windows Phone é um sistema operacional da *Microsoft* para *smartphones*. É o sucessor do extinto *Windows Mobile* (ZIEGLER, 2010). Lançado em outubro de 2007 conta com cerca de 1,1% do mercado mundial, segundo dados de ultimo trimestre de 2015 (GARTNER, 2016).

O *Windows Phone* se difere de seus concorrentes pela capacidade de integração com outros produtos da *Microsoft*, como Bing, *Xbox*, e o próprio sistema operacional *Windows* para *desktops* (BUCHANAN, 2010).

2.3.2 iOS

Antigamente conhecido como *iPhone OS*, *iOS* é um sistema operacional da *Apple Inc.* Possui a segunda maior parcela do mercado de *smartphones* do mundo, com 17,7% em 2015 (GARTNER, 2016). Em contraponto a possuir a segunda maior

parcela do mercado, iOS é o sistema operacional para *smartphones* que produz mais lucro, devido a agressiva competição de preço entre os fabricantes de celulares baseados em *Android* (LUNDEN, 2015).

Lançado em janeiro de 2007, introduziu inúmeros conceitos de *design* que foram amplamente adotados por outras plataformas de *smartphone*, como a capacidade de suportar inúmeros toques na tela de seu dispositivo de uma única vez (VOGELSTEIN, 2013).

2.3.3 Android

Android é um sistema operacional para *smartphones* desenvolvida pela *Google*. É uma plataforma *open source* com componentes proprietários, incluindo suporte para todos os serviços do *Google* (AMADEO, 2013).

Seu lançamento foi em outubro de 2008 e em 2015 possuía 80,7% do mercado mundial (GARTNER, 2016). Em 2014 a companhia revelou que possuem mais de um bilhão de usuários ativos mensalmente, comparando com os 538 milhões em junho de 2013 (KAHN, 2014).

2.4 LOJAS VIRTUAIS

No ano de 2008 a *Apple* inovou o mundo com o conceito de loja virtual nativa do próprio sistema operacional, onde até então a distribuição de aplicativos era feito por meio de terceiros. O conceito deu tão certo porque permitiu que desenvolvedores tivessem um canal direto com os usuários, que outras plataformas seguiram a mesma ideia adicionando lojas nativas para seus usuários poderem de maneira facilitada procurar e pesquisar suas necessidades.

Quando lançada, a *AppStore* da *Apple* possuía cerca de 500 aplicações disponíveis, sendo 75% delas paga (RICKER, 2008). Nos três primeiros dias de lançamento, 10 milhões de aplicativos foram baixados (APPLE, 2008). Em janeiro de

2015, a *AppStore* possuía mais de 1.4 milhões de aplicações, sendo essas coletivamente tendo sido baixadas 100 bilhões de vezes (INGRAHAM, 2015).

A *Google Play Store*, originalmente *Android Market*, é a versão da loja virtual do *Android*. Além de permitir a visualização e download de aplicativos publicados por meio de sua plataforma, a *Google Play* oferece também como loja de mídia digital, tendo musicas, livros, filmes e programas de televisão (CALLAHAM, 2015).

Inaugurada em março de 2012, *Google Play* foi o resultado da fusão do *Android Market*, *Google Music* e *Google eBookstore*. Em 2013 havia pouco mais de um milhão de aplicações na loja, sendo estes baixados mais de 50 bilhões de vezes (VICTOR, 2013).

2.5 TIPOS DE DESENVOLVIMENTO

A principal maneira de desenvolvimento móvel é a nativa em sua respectiva plataforma, é necessária então a replicação do código para quantas plataformas quiser alcançar, tornando alto o custo de desenvolvimento (CHARLAND; LEROUX, 2011). Porém ao mesmo tempo em que há garantia de a aplicação ter acesso a todo o *hardware* do dispositivo, obtendo-se assim um desempenho fluído e limpo no uso, existe também o ponto de sua manutenção ser extremamente custosa. Ocorrem situações em que o desenvolvedor precisa manter inúmeras versões para cada tipo de dispositivo móvel (MRC, N/I).

O desenvolvimento de plataformas cruzadas se destaca então pela possibilidade de reutilização de código para várias plataformas, ou mesmo a capacidade de escrever para elas fazendo uso da mesma linguagem, diminuindo assim o custo de manter desenvolvedores específicos para cada linguagem e plataforma. Isso sem perder o desempenho e capacidade de aplicações nativas pois ao menos três das quatro tecnologias estudadas neste projeto conseguem converter 100% do código escrito para código nativo.

Dentro do contexto de plataformas cruzadas existe também o desenvolvimento híbrido de aplicações. Aplicação híbrida é um misto de nativo com *web*. Codifica-se fazendo uso de HTML, CSS e *Javascript*, como uma aplicação *web* comum. O resultado é atribuído a um navegador nativo de cada dispositivo

permitindo assim o acesso a funções nativas do sistema como câmera, por exemplo. Seu desenvolvimento é simples comparado ao outros já citados, e sua grande característica é o fato de ser necessário escrever o código uma única vez para poder atingir diversas plataformas (MRC, N/I).

2.6 PLATAFORMAS CRUZADAS PARA DESENVOLVIMENTO DE APLICAÇÕES

Neste capítulo serão comentados os principais meios disponíveis no mercado para o tipo de desenvolvimento foco do trabalho.

2.6.1 Xamarin

Xamarin é uma companhia de *software* criada em 2011 no estado da Califórnia nos Estados Unidos (BINSTOCK, 2011). Com base na linguagem de programação C#, desenvolvedores podem usar o Xamarin para escrever aplicações para as plataformas *Android*, *iOS* e *Windows Phone*, podendo compartilhar códigos entre elas (XAMARIN, 2014). Possuem atualmente, segundo dados de fevereiro de 2016, mais de um milhão e trezentos mil desenvolvedores registrados tendo empregados em quatorze países diferentes.

Com Xamarin é possível fazer uso de 100% das APIs (*Application Programming Interface*) nativas da plataforma selecionada assegurando poder total para o desenvolvedor. Qualquer coisa escrita em *Objective-C* e *Java* podem ser feita em C# com Xamarin (XAMARIN, 2016).

É possível também fazer uso de bibliotecas *.NET* no aplicativo, ou mesmo facilmente consumir bibliotecas *Objective-C* e *Java*. Desde a *interface* até a lógica de negócio é possível ser escrita 100% em C# (XAMARIN, 2016).

Usando a mesma linguagem, APIs e estruturas de arquivos são possíveis compartilhar cerca de 75% do mesmo código através de todas as plataformas de desenvolvimento. Existe ainda um produto dentro do Xamarin chamado Xamarin Forms, onde por meio dele é possível reutilizar até 100% do código escrito para todas as plataformas (XAMARIN, 2016).

As ferramentas e meios fornecidos pelo Xamarin ganharam diversos prêmios mundo a fora desde sua criação, e como clientes eles possuem nome de peso como: *Microsoft, Slack, GitHub e Kellogg's*.

2.6.2 Apache Cordova

Apache Cordova (antigo PhoneGap), é um *framework* de código aberto para desenvolvimento de aplicações *mobile*. Originalmente criado pela Nitobi Softwares, *Apache Cordova* foi adquirido pela *Adobe Systems* em 2011 (ADOBE, 2011). Ele permite o uso de tecnologias *web* como HTML5, CSS3 e *JavaScript* para desenvolvimento multi-plataforma de aplicações.

Estas aplicações são conhecidas como aplicações híbridas, pois são construídas de maneira parecida que um *site* da *internet*. Porém, ao invés de focarem um *browser* do celular, elas fazem uso da *WebView* presente em cada plataforma. Isso permite que eles tenham acesso e capacidades de funções nativas do sistema (BRISTOWE, 2015).

Uma *WebView* não é muito mais que uma aba ou um *iframe* igual ao dos *browsers* existentes. Simplesmente pode-se dizer que aplicações híbridas são *sites* que funcionam em modo de tela cheia, mas que possuem acesso a funções nativas do sistema, como câmera, GPS e outros (LOOPER, 2015).

Como uma *WebView* é nativa de cada plataforma e a aplicação funciona em cima dela, é possível reutilizar 100% do código escrito, pois apenas a *WebView* muda.

Por fazer uso de componentes HTML, existem ferramentas visuais para criação de telas e uso de componentes simples. Facilitando o uso para qualquer pessoa, mesmo sem ter conhecimento específico sobre programação.

É possível também por meio do *Apache Cordova* converter aplicações para diversas outras plataformas, como: *FirefoxOS, LG Web OS, Ubuntu* e até *BlackBerry*.

Vale notar também que grandes organizações como *Basecamp, Instagram* e *Yelp* estão fazendo uso de aplicações híbridas e tendo muito sucesso com elas, não deixando a desejar em visual, *performance* e resultados.

2.6.3 Unity

Unity é um motor de jogos multi-plataforma criado pela *Unity Technologies* em 2005, e, desde então suporta mais de quinze plataformas diferentes de desenvolvimento (UNITY, 2015) (TAKAHASHI, 2014).

Atualmente possui 45% do mercado de jogos do mundo, possuindo também 47% dos desenvolvedores de jogos e tendo chegado à marca histórica de 600 milhões de jogadores que jogaram algo feito pelo motor Unity (UNITY, 2015).

Com o foco no desenvolvimento de jogos, Unity foi escrita em C e C++, mas faz uso da linguagem C# para criação das aplicações (AMAZONAS, 2014).

Funcionando dentro da máquina virtual Mono, a mesma em que o Xamarin se baseia, é possível criar qualquer tipo de jogo e convertê-lo ao final para mais de quinze plataformas diferentes. Porém mesmo o foco deste motor ser o de jogos, é possível também utilizá-lo para criar aplicações mais usuais de celulares usando 100% o mesmo código e convertendo a aplicação para as plataformas desejadas em apenas um clique.

2.6.4 Delphi 10

Delphi é uma IDE (*Integrated Development Environment*) para desenvolvimento de aplicações *desktop*, *mobile*, *web* e *consoles* (EMBARCADERO, 2015). Originalmente criada pela Borland para ser uma ferramenta rápida de desenvolvimento para *Windows* em 1995 e derivada do *Turbo Pascal*.

Delphi faz uso do pascal, que por sua vez é uma linguagem fortemente tipada e de alto nível, feita com a intenção de ter um uso simples. O Delphi possui uma IDE integrada a ele, que faz uso de *drag and drop* (arrastar e soltar) para lidar com os elementos visuais em seus formulários.

Na última versão estável da IDE Delphi, por meio da FireUI, é possível criar aplicativos para celulares focando em várias plataformas utilizando também 100% do

mesmo código escrito e criando telas e formulários com um simples arrastar e soltar de componentes (EMBARCADERO, 2015).

Os compiladores do Delphi usam seu próprio dialeto *Object Pascal* do *Pascal* e assim conseguem gerar os códigos para todas as plataformas suportadas.

Seus principais clientes variam desde a área médica, informática e comunicação.

3 MATERIAL E MÉTODOS

Este capítulo tem como objetivo descrever as ferramentas, *frameworks* e tecnologias utilizadas para o desenvolvimento do protótipo de aplicação.

3.1 FERRAMENTAS UTILIZADAS

Foram utilizadas as seguintes ferramentas e tecnologias para o desenvolvimento do protótipo de aplicação:

- Sistema Operacional Windows 10;
- Linguagem C#;
- Visual Studio 2013;
- Visual Studio 2015;
- *Entity Frameworks*;
- Web API;
- Sistema Gerenciador de Banco de Dados MySQL;
- Apache Cordova;
- *Plugins* para Apache Cordova;
- *IonicFramework*;
- AngularJS;
- JavaScript;
- HTML.
- *Smartphone* Samsung Galaxy S6;

O sistema operacional *Windows 10* foi escolhido, pois é o ambiente onde é possível fazer uso da IDE (*Integrated Development Environment*) Visual Studio, que por sua vez é um pacote de programas da *Microsoft* para desenvolvimento de programas especialmente dedicados à linguagem *Visual Basic*, C, C++ e C#. Permite também o desenvolvimento de *Web Sites*, aplicações *Web* e *Web services*.

A versão 2013 do Visual Studio foi escolhida para ser feita a parte do servidor da aplicação. É uma versão estável e conta com compatibilidade para inúmeros plug-ins, *frameworks* e bibliotecas que ajudam no desenvolvimento.

O Visual Studio é a principal IDE para desenvolvimento da linguagem C# (lê-se: “cêxarpe”) junto ao *.NetFramework* para compilação de projetos. C# é uma linguagem de programação interpretada multi-paradigma fortemente tipada. Sua sintaxe orientada a objetos deriva-se da linguagem C++, porém há influências de outras linguagens como *Object Pascal* e *Java*.

O *Entity Frameworks*, ou *ADO.NET Entity Frameworks* é um *framework open source* que permite fazer o mapeamento de objetos relacionais. É uma das principais ferramentas de persistência de dados para a plataforma *.NET*, permitindo aumentar muito a produtividade no desenvolvimento de aplicações persistentes. Permite também uma comunicação fácil e simples com o banco de dados MySQL, que por sua vez foi escolhido pela fácil manutenção. Para este protótipo foi usado a versão 5 do *Entity Frameworks* no servidor.

A API (*application programming interface*) *Web API* pode ser utilizada tanto para um servidor quanto para um navegador. Na parte do servidor, ela possui uma interface programática que consiste de um ou mais *endpoints* expostos publicamente para um sistema que pedido-resposta. Normalmente expressado por JSON (*JavaScript Object Notation*) ou XML (*extensible markup language*), é comumente utilizado por meio de um servidor *web* baseado em HTTP (*Hypertext Transfer Protocol*). É por meio do *Web API* que possível a comunicação entre a aplicação e o servidor. Foi feito uso da versão 4.0.

Por conveniência de como o *Visual Studio 2015* organiza os arquivos de um projeto *Apache Cordova* ele foi escolhido para ser desenvolvido o protótipo de aplicação. *Apache Cordova* possui dependência do *Node.js* para funcionar, a versão do *Node.js* usada foi a 0.12.9 e do *Apache Cordova* 5.4.1.

Sendo um *framework* popular para aplicações *mobile*, o *Apache Cordova* permite que desenvolvedores produzam aplicações *mobile* usando CSS3, HTML5 e JavaScript. O código escrito é atribuído a uma *WebView* nativa da plataforma desejada, possuindo assim a capacidade de acessar e controlar funções nativas do *hardware*.

Para ser possível o acesso a capacidades nativas de cada celular e plataforma, o *Apache Cordova* faz uso de *plug-ins* que provem uma *interface*

Javascript para essas funções. Eles permitem então a aplicação um poder sobre o celular maior que o que seria disponível apenas em uma aplicação *Web*.

Neste protótipo foram utilizados os seguintes *plug-ins Apache Cordova*:

- Cordova-Plugin-Device, versão 1.1.1;
- Cordova-Plugin-Geolocation, versão 2.1.1;
- Cordova-Plugin-Splashscreen, versão 3.2.1;
- Cordova-Plugin-Statusbar, versão 2.1.2;
- Cordova-Plugin-Whitelist, versão 1.2.1;
- Ionic-Plugin-Keyboard, versão 2.0.1.

Para a parte de aparência e funcionalidades da aplicação, foi escolhido o *Ionic Framework*. Seu propósito maior é com as interações de *interface*, dando a aplicação uma aparência nativa. Ele faz uso do *AngularJS*, o que permite uma produtividade superior a simplesmente usar apenas *JavaScript*. Foi feito uso da versão 1.7.14 do *Ionic Framework*, e 0.7.0 de sua biblioteca.

Por fim, usou-se o celular Samsung Galaxy S6 para testar e utilizar o protótipo de aplicação desenvolvido.

3.2 ESTUDO DAS PLATAFORMAS

A fim de investigar os aspectos básicos que envolvem o desenvolvimento de aplicativos baseados em plataformas cruzadas, foram criados projetos em cada plataforma selecionada.

As plataformas selecionadas para o estudo foram: Apache Cordova, Embarcadero Delphi 10 e Microsoft Unity.

Os aspectos investigados na criação dos projetos demonstrativos nas plataformas selecionadas foram avaliar a complexidade em criar um projeto para a plataforma específica, criar um botão e adicionar a uma ação de clique a ele.

3.3 ESTUDO DO PROTÓTIPO

Com o objetivo de aprofundar o estudo no desenvolvimento baseado em plataforma cruzada, foi desenvolvido um protótipo de aplicação baseada em *Apache Cordova*, com o escopo de controle de viagens para funcionários da UTFPR Medianeira.

Atualmente quando um servidor da instituição precisa se deslocar em uma viagem para outra cidade, é necessário que a mesma seja devidamente registrada para que possa assim haver um controle e contabilização dos dados informados. Esse registro é realizado por meio de uma ficha onde as informações são preenchidas, como: data de saída, data de retorno, número de abastecimentos, quilometragem inicial e final do veículo e etc. O processo realizado não é o ideal pois não há como emitir relatórios, produzir gráficos ou visualizar de maneira simples informações relevantes quanto as viagens.

Com os conhecimentos estudados e obtidos durante o desenvolvimento deste trabalho de conclusão produziu-se então uma solução móvel que atende os requisitos mínimos levantados para automatizar o processo, torna-lo fácil, dinâmico e que pode ser realizado de qualquer lugar, desde que haja conexão com a *Internet*.

3.3.1 Diagrama De Caso de Uso

Representado pela Figura 1, o diagrama de caso de uso mostra o protótipo com suas principais funcionalidades. Sendo estes: manter pessoas, manter veículos e manter viagens.

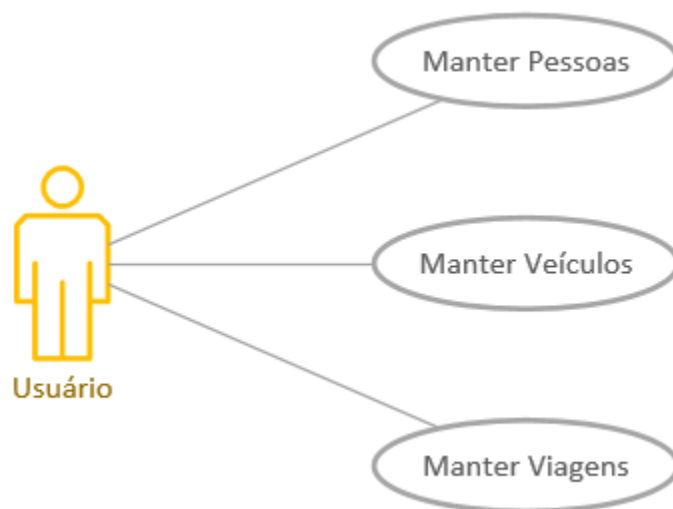


Figura 1 – Diagrama de caso de uso.
Fonte: Autoria própria.

A descrição detalhada dos casos de uso estão se encontram no Apêndice A: manter pessoas (Quadro 1), manter veículos (Quadro 2), manter viagens (Quadro 3).

3.3.1 Diagramas De Sequência

A Figura 2 e Figura 3 apresentam o fluxo da principal função do sistema: gravar e finalizar viagem.

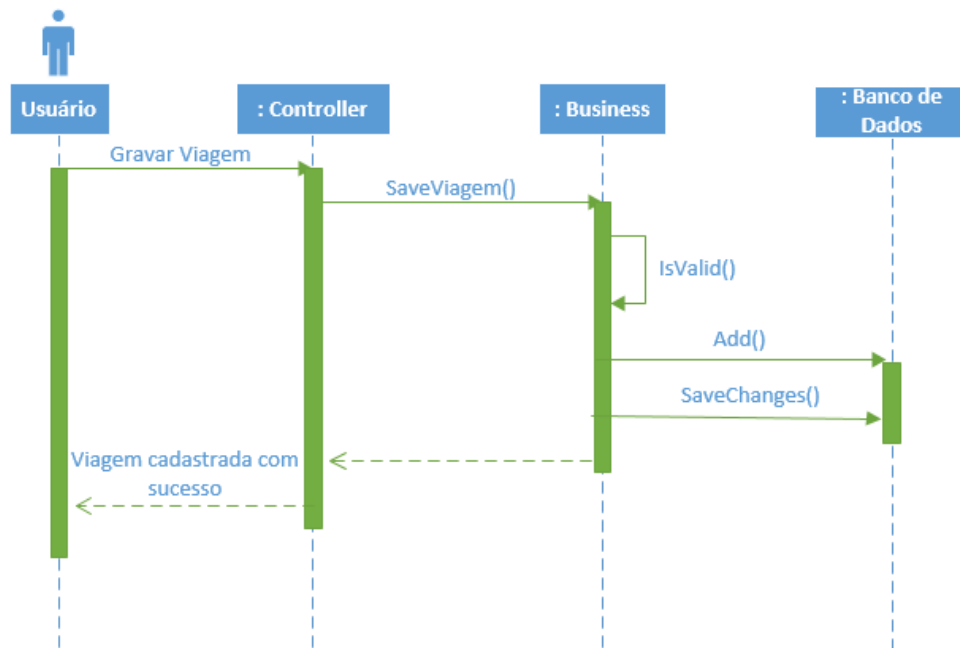


Figura 2 – Diagrama de seqüência (cadastro viagem).
Fonte: Autoria própria.

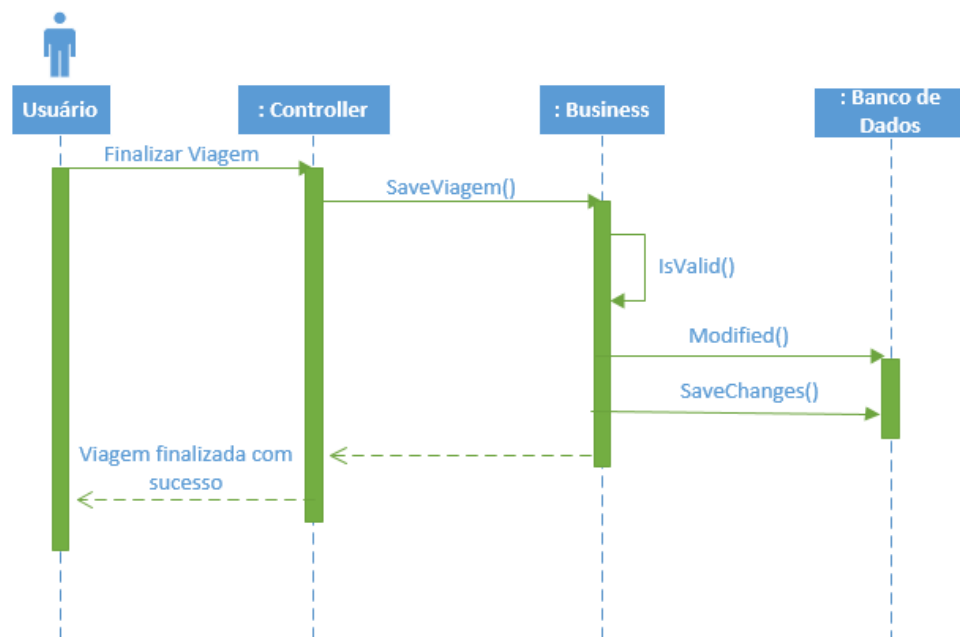


Figura 3 – Diagrama de sequencia (finalizar viagem).
Fonte: Autoria própria.

3.3.2 Diagrama de Atividade

O diagrama de atividades representado pela Figura 4 exibe as ações necessárias para a gravação de uma viagem.

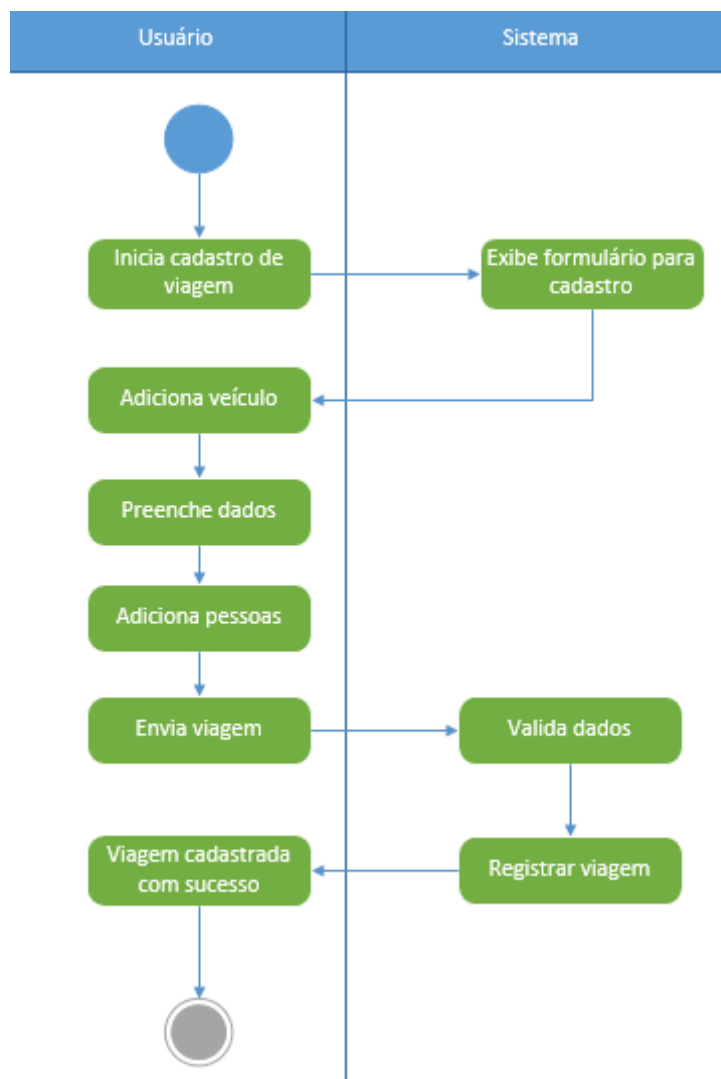


Figura 4 – Diagrama de atividades.
Fonte: Autoria própria.

3.3.3 Diagrama de Classe

A Figura 5 expõe o diagrama de classes utilizado pelo protótipo desenvolvido.

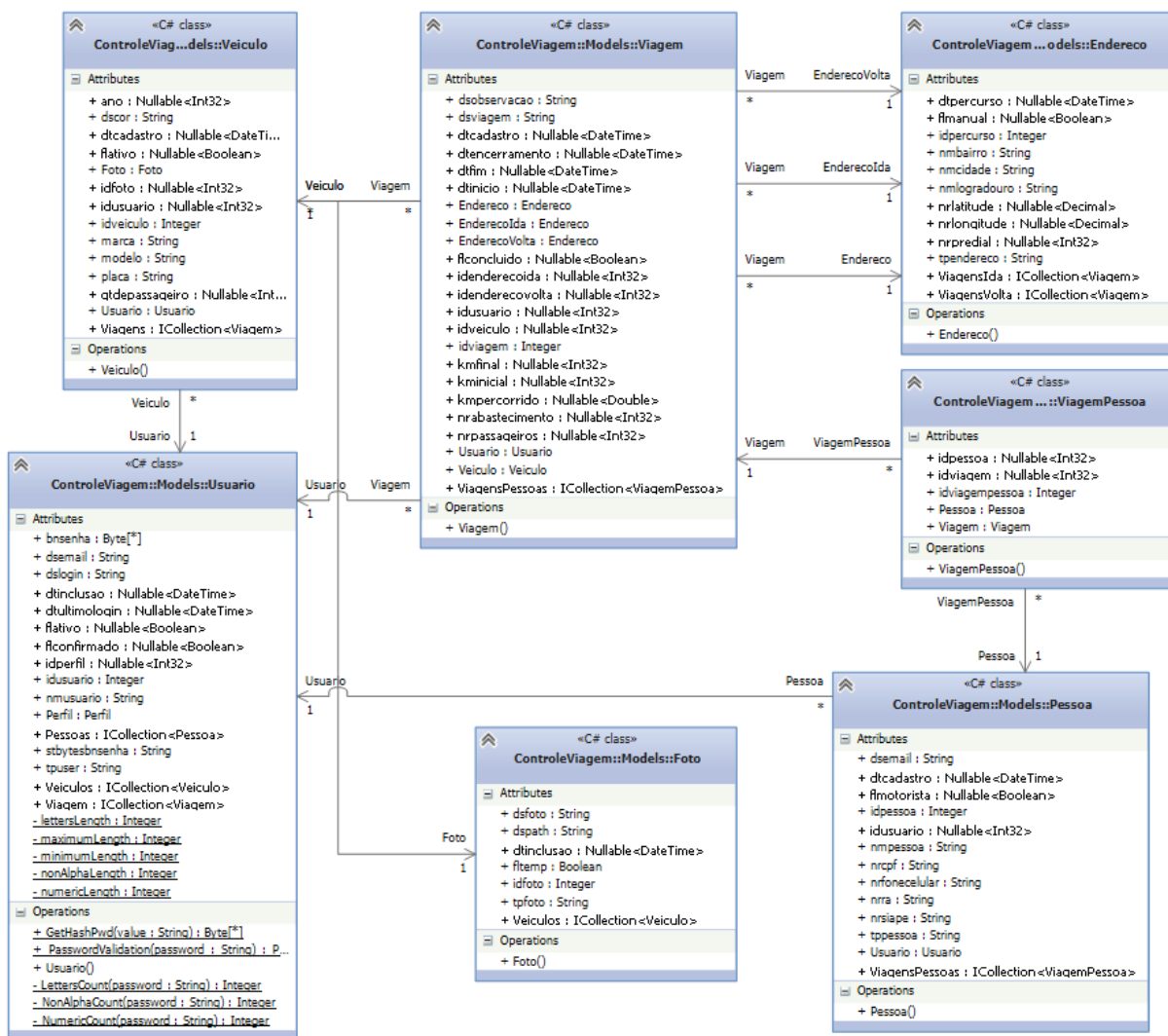


Figura 5 – Diagrama de classes.
 Fonte: Autoria própria.

3.3.4 Modelo De Dados– MER

O modelo conceitual utilizado para definir as entidades e como elas se relacionam entre si está representado pela Figura 6.

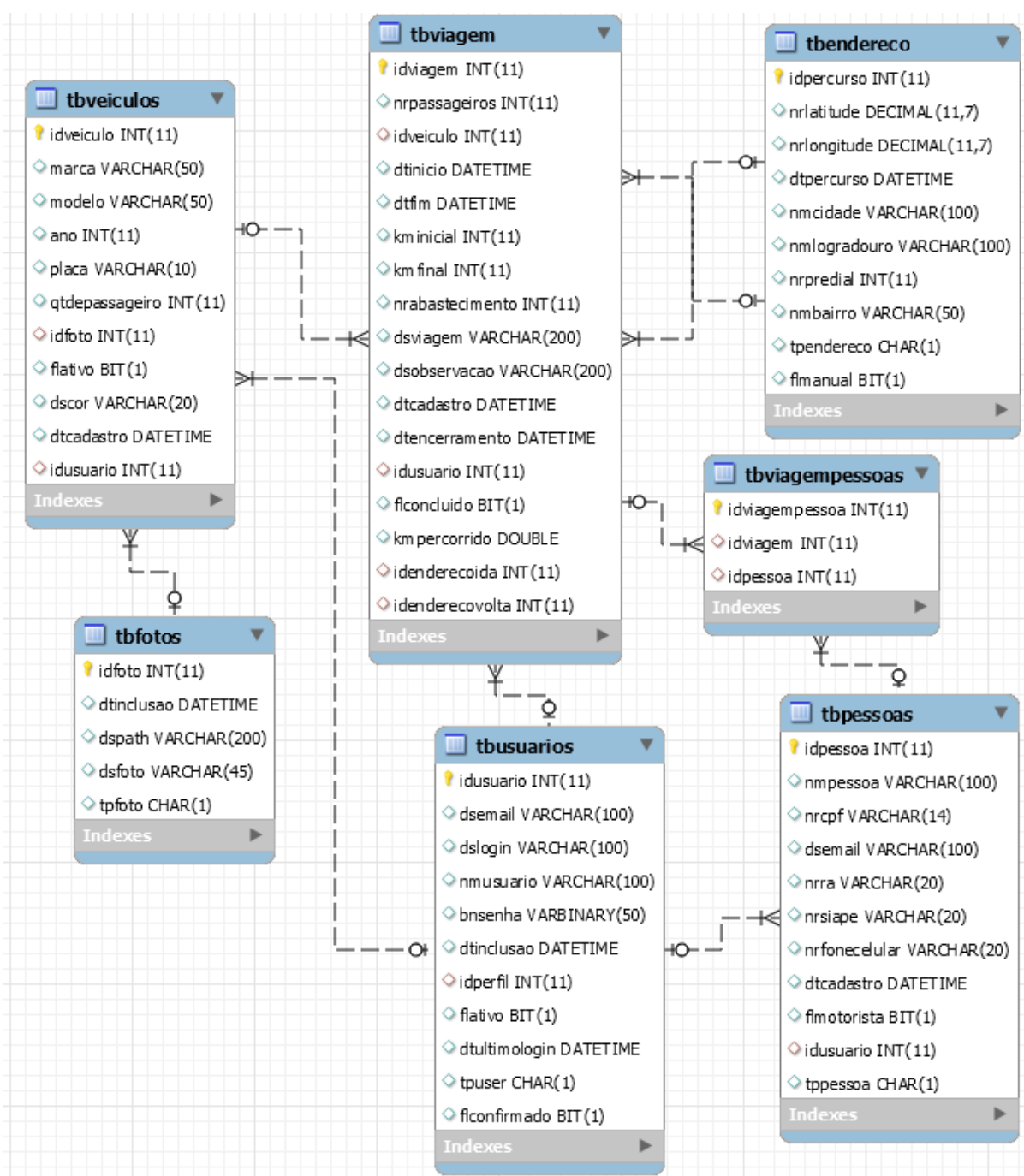


Figura 6 – MER do protótipo.

Fonte: Autoria própria.

3.3.5 Diagrama de Telas

A Figura 7 exemplifica o fluxo de navegação das telas do sistema.

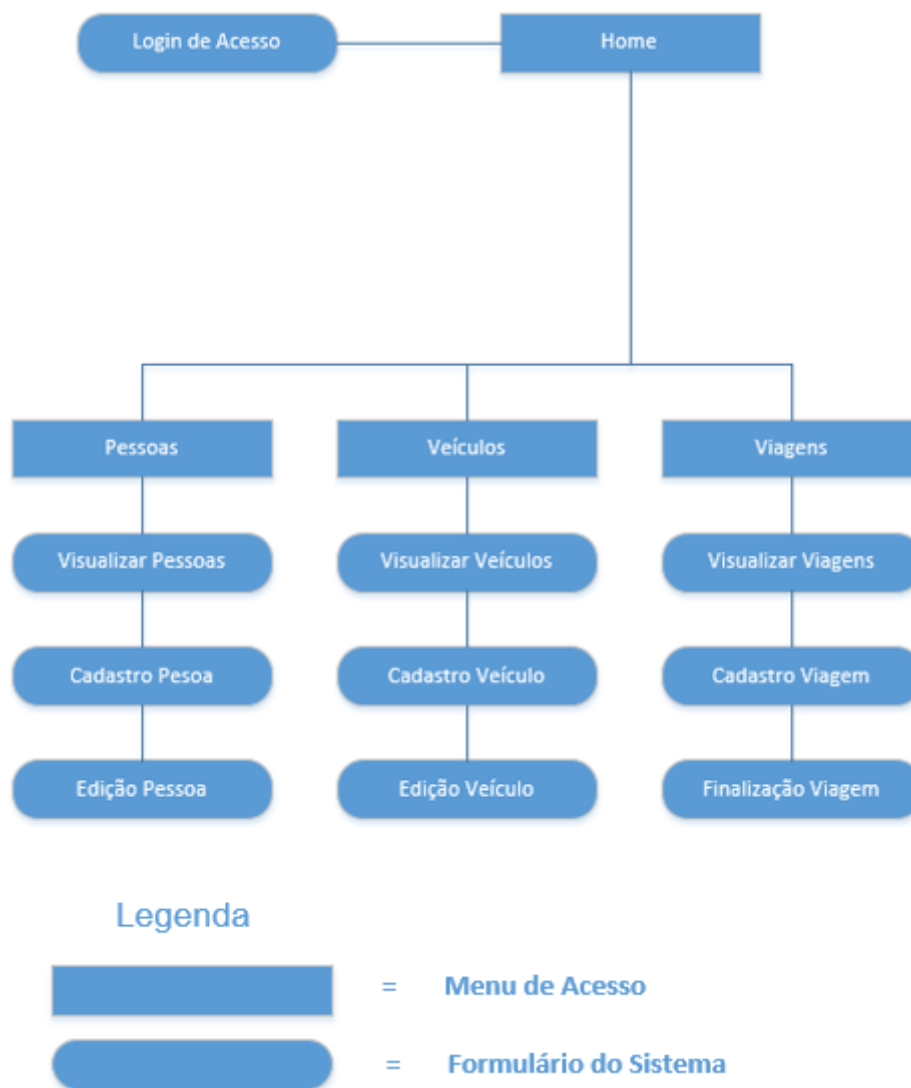


Figura 7 – Diagrama de telas.
Fonte: Autoria própria.

4 RESULTADOS E DISCUSSÃO

4.1 INVESTIGAÇÃO DAS PLATAFORMAS

Como resultados da investigação quanto às ferramentas e linguagens utilizadas para o desenvolvimento de aplicativos em plataformas cruzadas, foram criados projetos em cada plataforma selecionada. Com a criação das demonstrações, teve-se como finalidade avaliar a complexidade em criar um projeto para a plataforma específica, criar um botão e adicionar a uma ação de clique a ele.

4.1.1 Projeto Demonstrativo Em Delphi 10

Para este exemplo foi utilizada a versão Embarcadero® Delphi 10.1 Berlin. Primeiramente depois de instaladas as devidas dependências para desenvolvimento *mobile* para plataforma *Android*, que será o alvo destes testes, clicou-se em: File -> New -> Multi-Device Application (Figura 8).

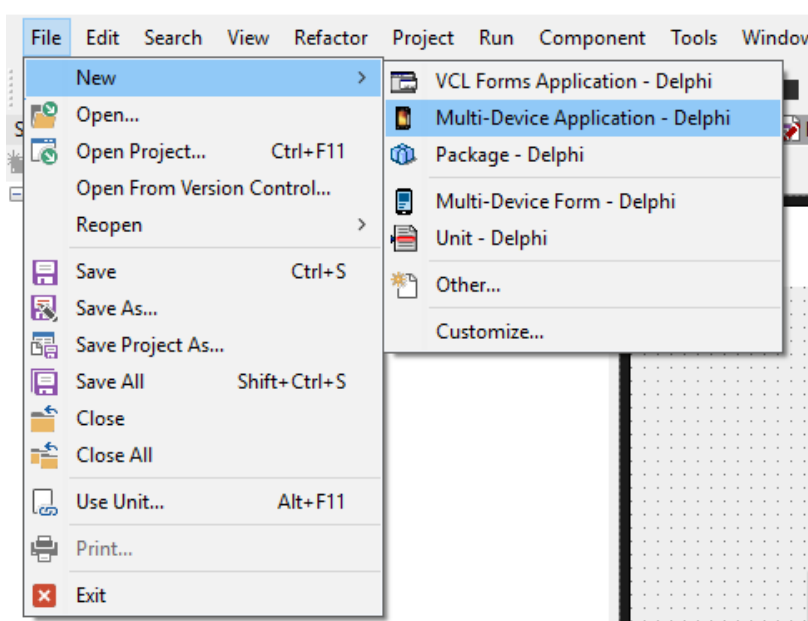


Figura 8 – Criação projeto Delphi 10.
Fonte: Autoria própria.

Em seguida várias opções de tipos de aplicações são oferecidas como base. Selecionou-se então a opção “*Header/Footer*” (Figura 9).

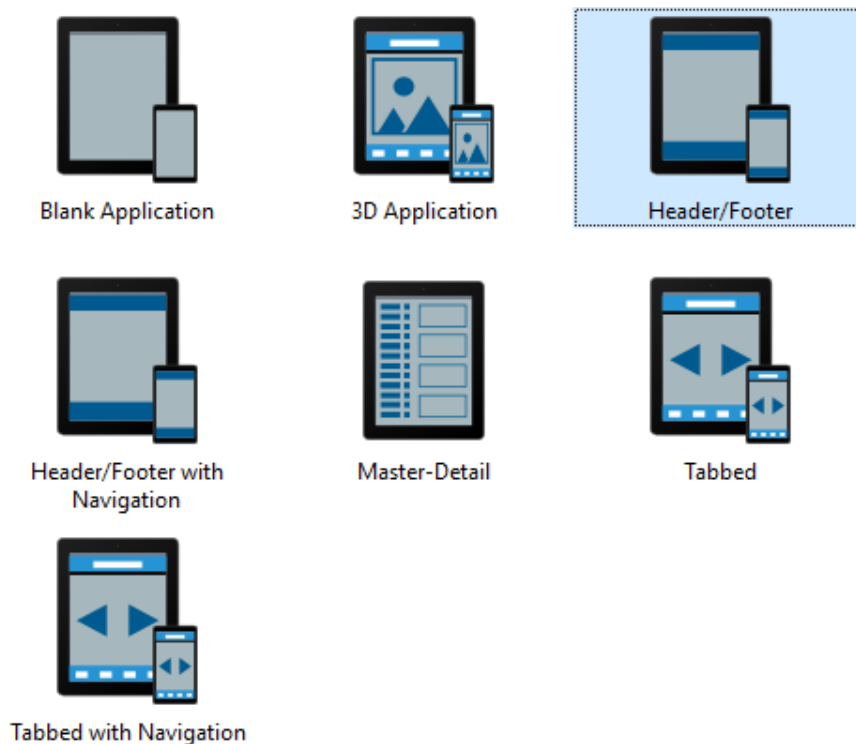


Figura 9 – Opções de aplicação Delphi 10.
Fonte: Autoria própria.

Uma interface visual então é mostrada, e nela, como se é comum da plataforma Delphi, existe a possibilidade de se trabalhar arrastando e soltando componentes na tela. Um botão então foi adicionado (Figura 10).

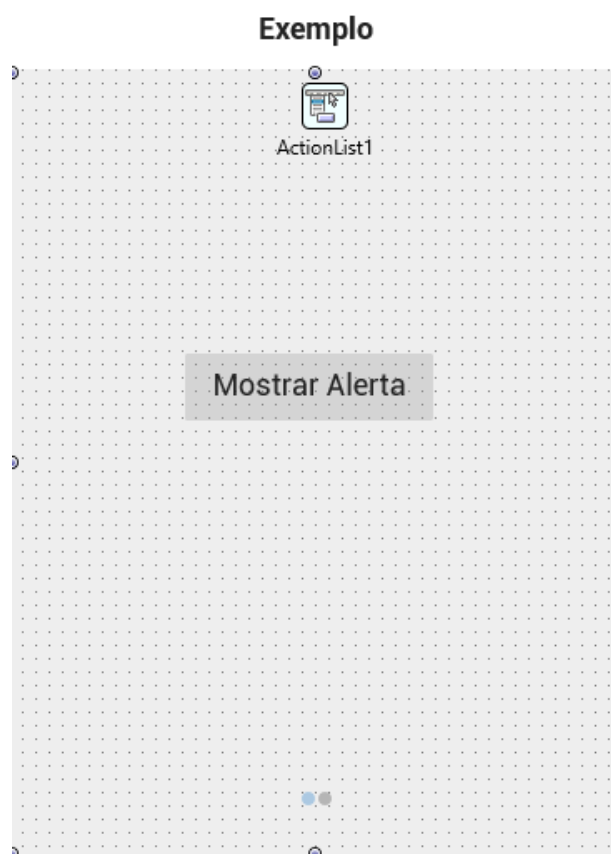


Figura 10 – Interface visual de componentes Delphi 10.
Fonte: Autoria própria.

Ao dar-se um duplo clique no botão, a tela é redirecionada para o método responsável pela ação deste evento. Neste caso um *ShowMessage()*, foi utilizado pois sua sintaxe é referente a linguagem Pascal (Figura 11).

```
procedure THeaderFooterwithNavigation.Button1Click(Sender: TObject);  
begin  
    ShowMessage('Olá UTFPR');  
end;
```

Figura 11 – Código de alerta Delphi 10.
Fonte: Autoria própria.

Por fim, para executar a aplicação criada em um *smartphone* é necessário apenas ligar o mesmo via cabo USB no computador. Selecionou-se então a plataforma desejada, no caso *Android* e clicou-se no botão “Run” (Figura 12).

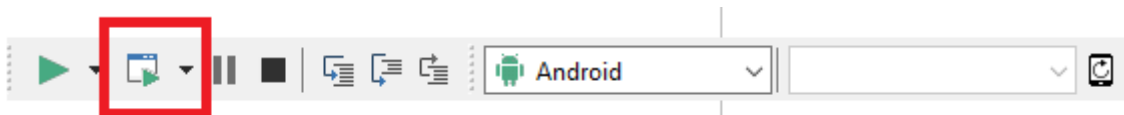


Figura 12 – Executar aplicação Delphi 10.
Fonte: Autoria própria.

Após a compilação do projeto, automaticamente a aplicação se inicializará no *smartphone*. Lembrando que mesmo a aplicação tendo sido escrita em Pascal, ela é convertida 100% para um código nativo, permitindo então ter-se uso de toda a capacidade e serviços do *hardware* e plataforma que deseja usar em um celular.

Representado pela Figura 13 e Figura 14, é mostrado então a demonstração em execução.

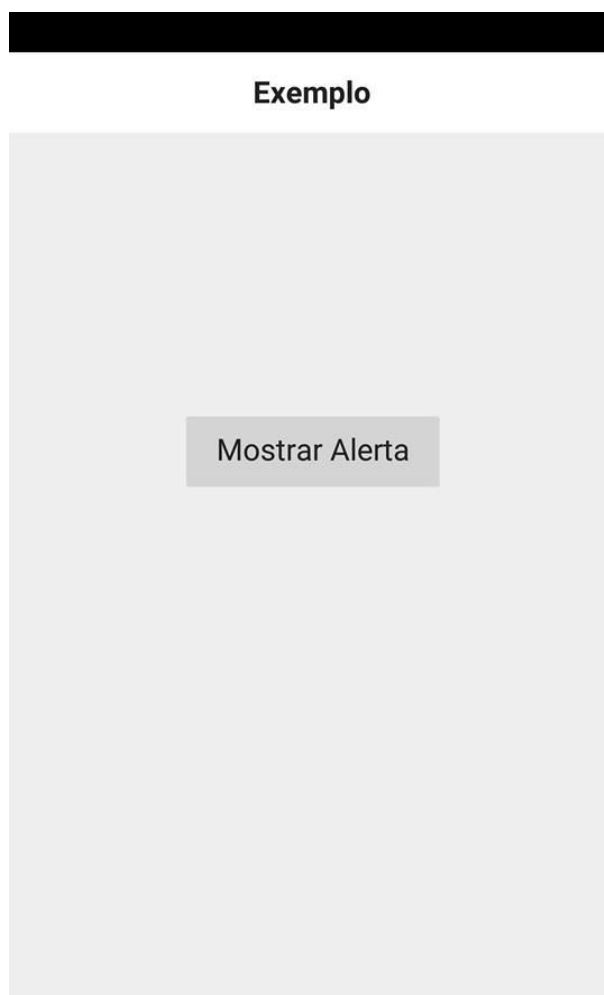


Figura 13 – Tela da aplicação Delphi 10.
Fonte: Autoria própria.



Figura 14 – Mensagem da aplicação Delphi 10.
Fonte: Autoria própria.

4.1.2 Projeto Demonstrativo Em Apache Cordova

Podem-se desenvolver aplicações inteiras usando apenas um editor de texto comum com *Apache Cordova*. Isso é possível, pois sua execução depende do *Node.js*. Quando completo o código, basta apenas por meio de linha de comando construir e passar a aplicação para o *smartphone* na plataforma desejada.

Outro meio de desenvolver aplicações com esta tecnologia é por meio de uma IDE. Este exemplo será desenvolvido usando *Visual Studio 2015*. Manualmente as dependências para o desenvolvimento podem ser baixadas, porém existe um pacote

já contendo todo o necessário para o *Visual Studio*, seu nome é “*Tools for Apache Cordova*”.

Após a instalação do pacote, para criar a aplicação é necessário dentro da IDE acessar o menu: *File -> New -> Project* (Figura 15).

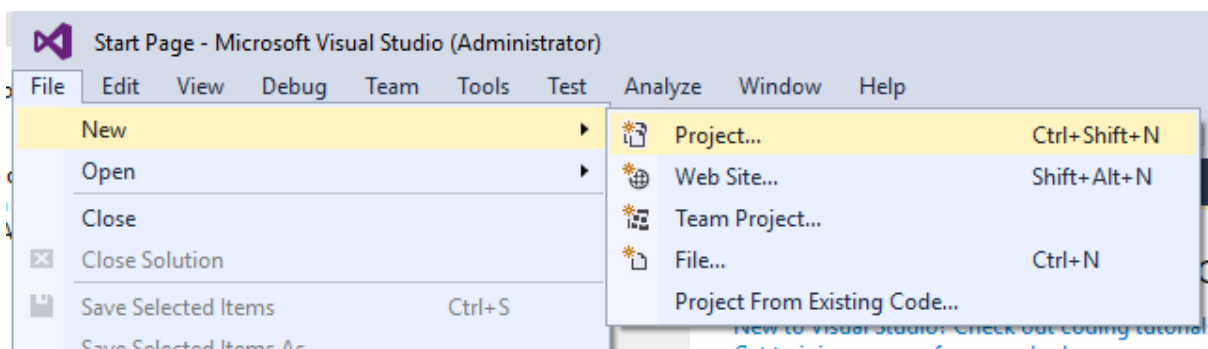


Figura 15 – Criação projeto Apache Cordova.
Fonte: Autoria própria.

Na sessão *JavaScript* selecionou-se um *BlankApp*, ou seja, um projeto em branco do tipo *Apache Cordova* e a criação foi confirmada (Figura 16).

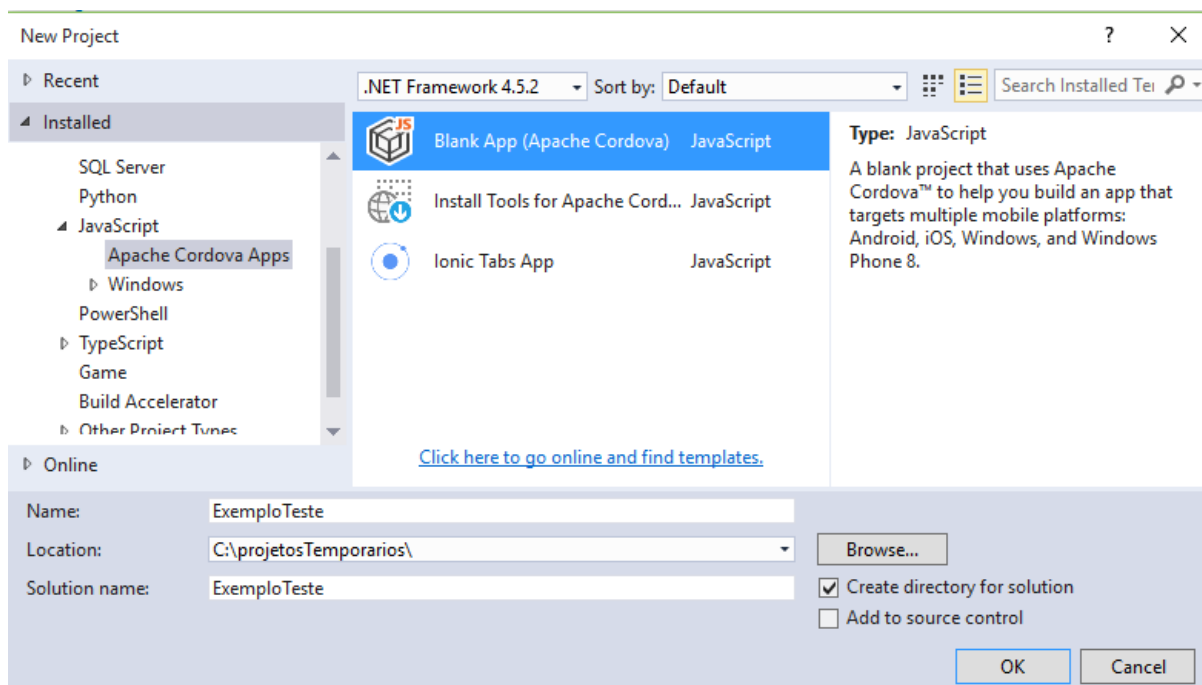


Figura 16 – Tipo de projeto Apache Cordova.
Fonte: Autoria própria.

A estrutura de arquivos criada é muito semelhante a um projeto *web* padrão do *Visual Studio* (Figura 17).

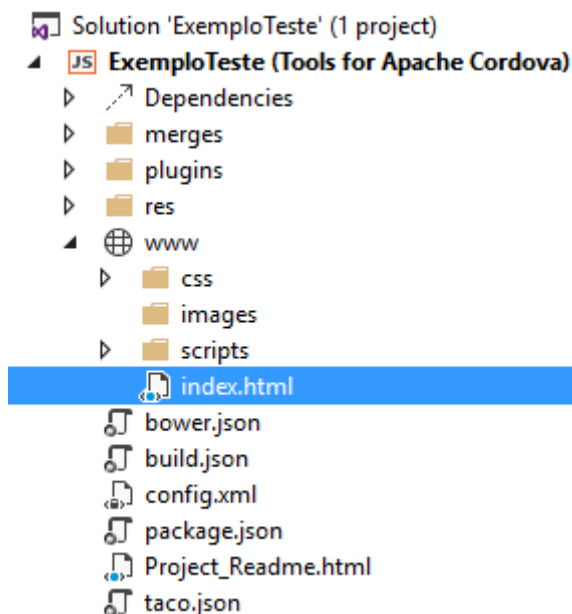


Figura 17 – Estrutura de projeto Apache Cordova.
Fonte: Autoria própria.

Acessando o arquivo “*index.html*”, pode-se notar que seu conteúdo é todo HTML, e é esse HTML que será convertido para de maneira híbrida funcionar na plataforma que deseja. Vale ressaltar que qualquer biblioteca ou *script* pode ser utilizado, como os famosos *jQuery* e *AngularJS*, por exemplo.

Foi adicionado então a *tag* HTML referente a um botão, e em sua ação foi adicionado o código para exibir um simples alerta com um texto (Figura 18).

```
<body>
  <p>Exemplo</p>
  <button type="button" onclick="alert('Olá UTFPR')">Mostrar Alerta</button>
</body>
```

Figura 18 – Exemplo código aplicação Apache Cordova.
Fonte: Autoria própria.

Para executar a aplicação no celular, é necessário apenas selecionar a plataforma e o destino, neste caso é “*Device*”, pois há como executar também em emuladores para testes (Figura 19).

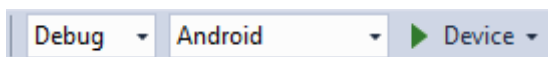


Figura 19 – Executar aplicação Apache Cordova.
Fonte: Autoria própria.

O resultado então é demonstrado pelas imagens seguintes (Figura 20, Figura 21).

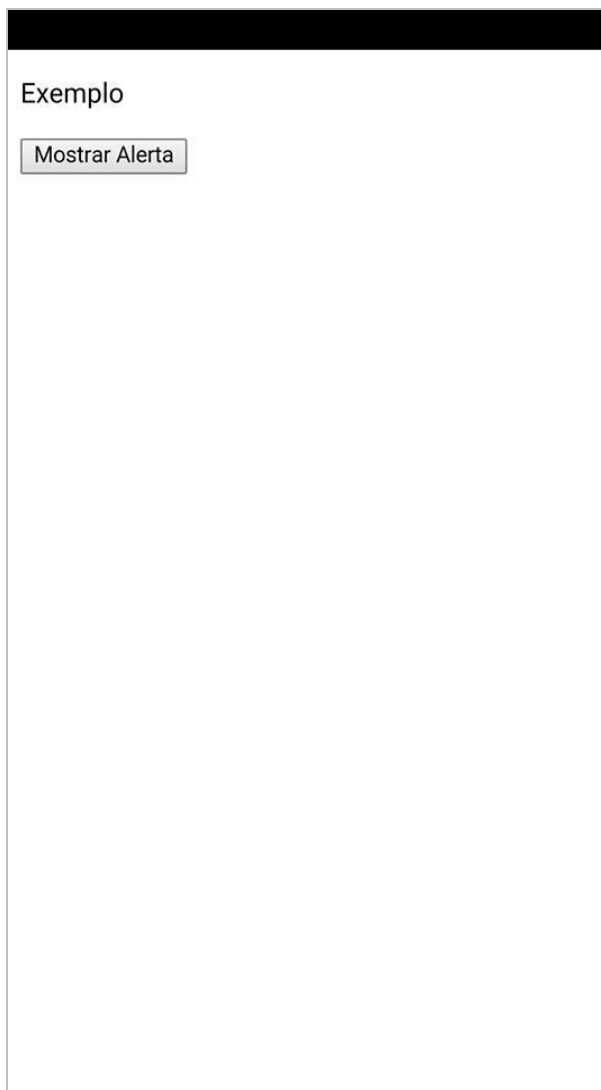


Figura 20 – Tela aplicação Apache Cordova.
Fonte: Autoria própria.

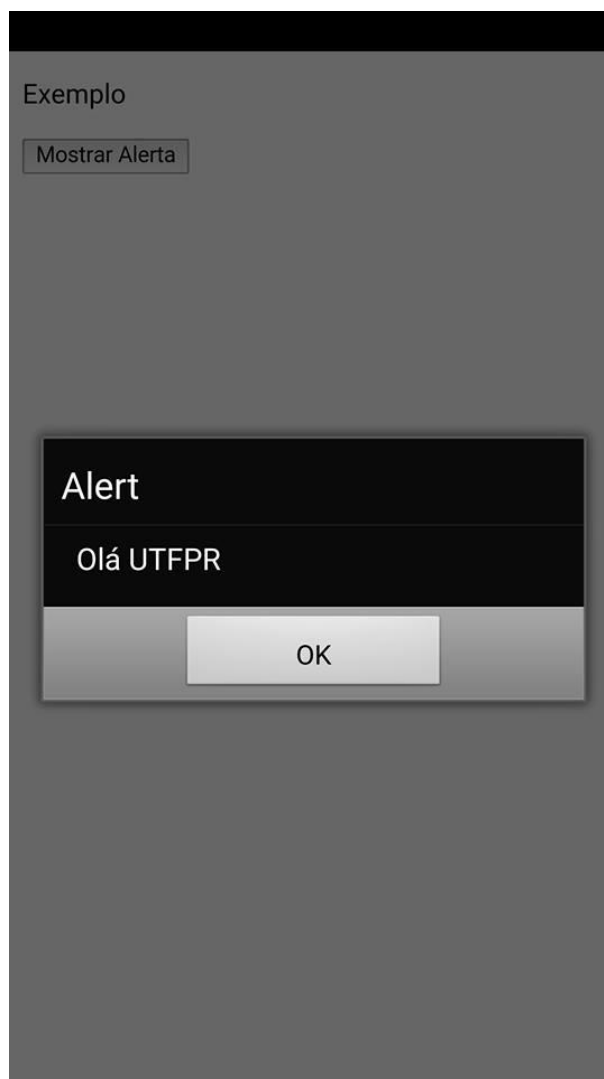


Figura 21 – Mensagem da aplicação Apache Cordova.
Fonte: Autoria própria.

4.1.3 Projeto Demonstrativo Em Unity

Para este teste a versão 5.3.1f1 do Unity foi utilizada. Iniciou-se um novo projeto clicando em *File -> New Project* (Figura 22).

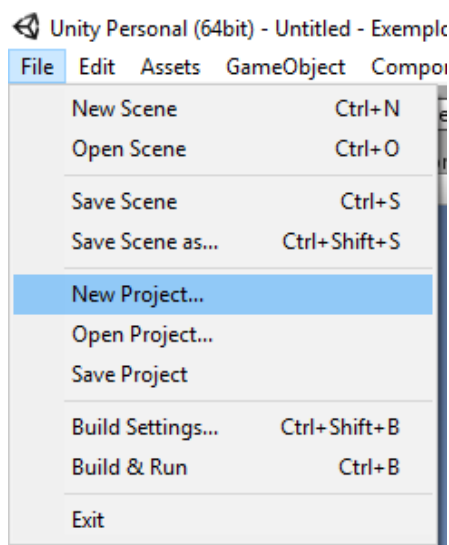


Figura 22 – Criação projeto Unity.
Fonte: Autorial própria.

Foi selecionado então a opção “2D” e informado um nome ao projeto (Figura 23).

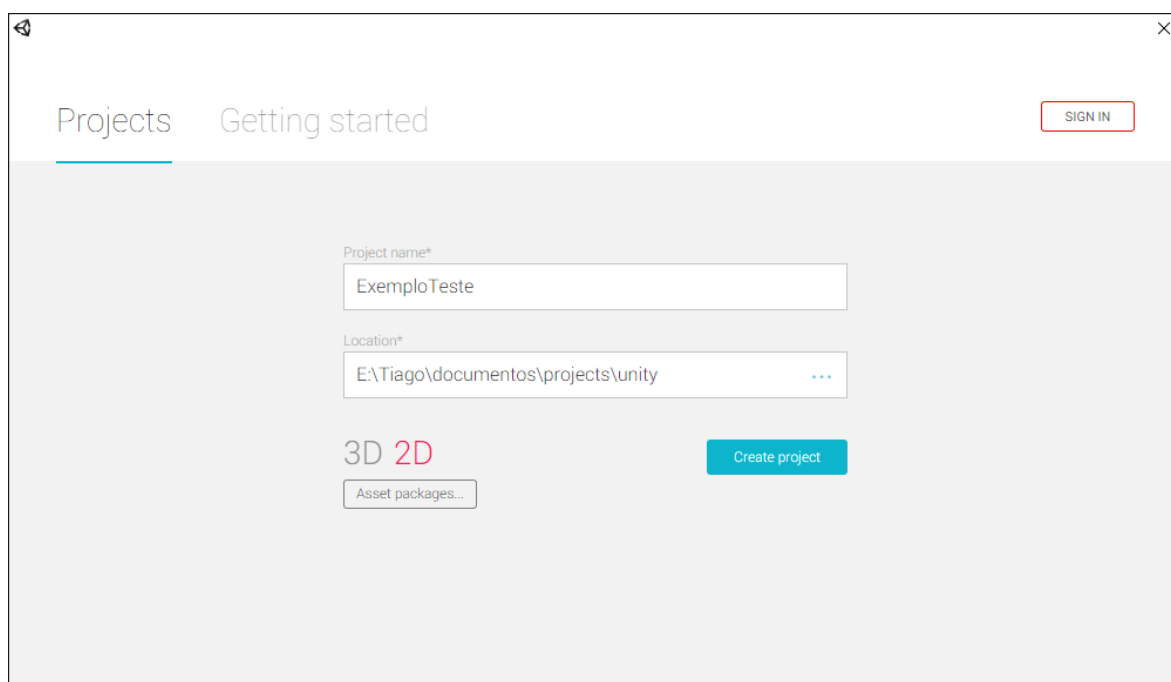


Figura 23 – Tipo do projeto Unity.
Fonte: Autorial própria.

Notou-se que na sessão *Hierarchy* alguns componentes vem criados automaticamente, como câmera e direção de luz, pois o foco principal do Unity é justamente a criação de jogos. Ainda na sessão *Hierarchy* foi clicado com o botão

direito em um lugar livre, selecionado a sessão UI e adicionado um *Button* (Figura 24).

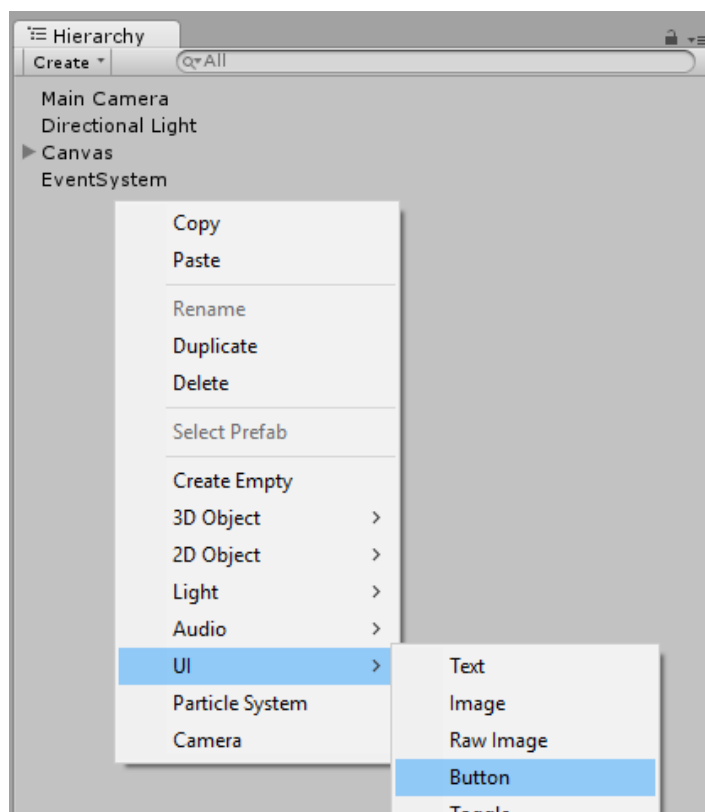


Figura 24 – Novo botão Unity.
Fonte: Autoria própria.

Dando prosseguimento a demonstração, um componente de texto foi adicionado, seguindo o padrão do passo anterior, porém selecionando-se *Text* (Figura 25).

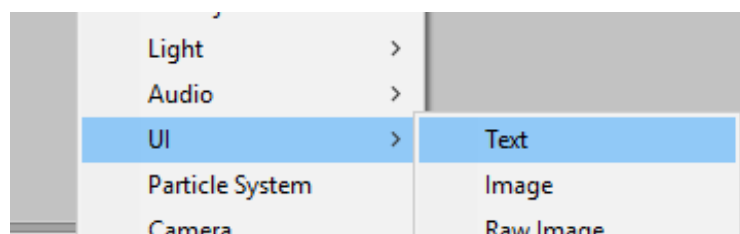


Figura 25 – Novo texto Unity.
Fonte: Autoria própria.

Selecionando o componente *Text*, removeu-se o texto adicionado automaticamente a ele em *Text (Script)*. Para ser atribuída uma ação ao botão

adicionado é necessário criar um *script* a ele. Na sessão *Assets* em *Project*, clicou-se com o botão direito e foi selecionado o menu: *Create -> C# Script* (Figura 28).

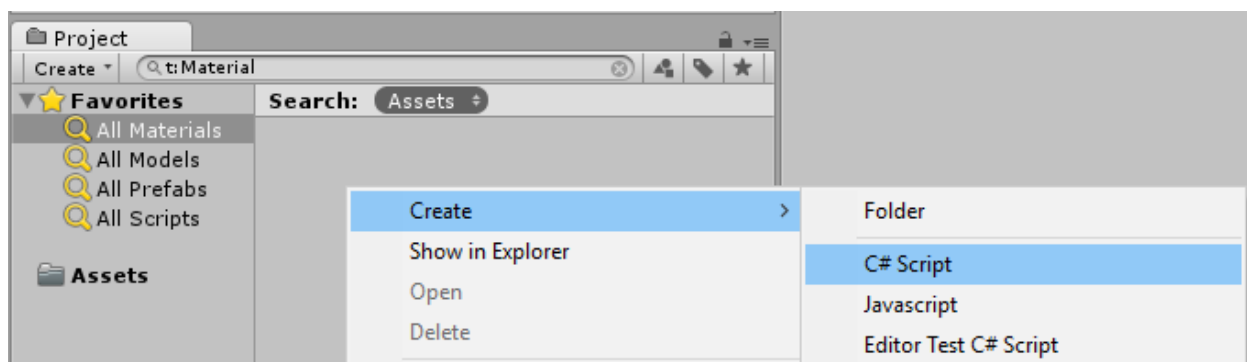


Figura 26 – Novo script Unity.
 Fonte: Autoria própria.

Abrindo o *script* criado, adicionou-se o método *ShowText()* como consta na Figura 27.

```
public void ShowText()
{
    textObj = GameObject.Find("Text").GetComponent<Text>();
    textObj.text = "Olá UTFPR";
    textObj.fontSize = 150;
    textObj.color = Color.white;
}
```

Figura 27 – Código botão Unity.
 Fonte: Autoria própria.

Talvez a parte mais complicada desta demonstração foi a atribuição da ação do método criado para o botão adicionado. Para isso é necessário arrastar o *script* criado para o componente *MainCamera* em *Hierarchy*. Em seguida foi selecionado o botão adicionado e em *On Click ()* dentro de *Inspector* arrastou-se o componente *Main Camera*. Uma caixa com opções apareceu e nela pode-se usar o *script* criado juntamente com o método adicionado a ele (Figura 28).

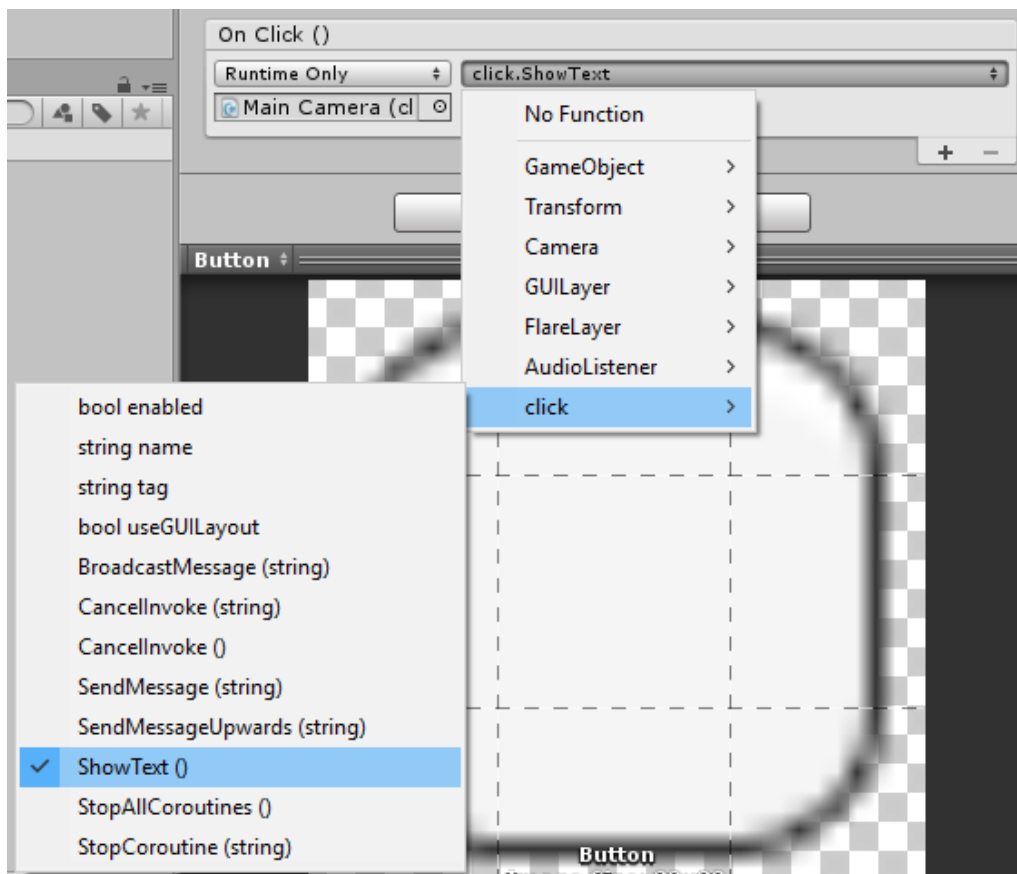


Figura 28 – Atribuição de evento para botão Unity.
Fonte: Autoria própria.

Ao clicar no botão um novo texto, cor e tamanho serão atribuídos ao componente de texto adicionado. Nesta demonstração não foi possível seguir o exemplo das demais em abrir um simples alerta, pois não foi encontrado de maneira simples um jeito de assim o fazer. Seria necessária a criação manual de uma janela de diálogo e funções referentes a ela, fugindo então do objetivo aqui proposto. Ficou evidente a dificuldade e complexidade de lidar com ações simples de aplicações cotidianas, pois como já mencionado, o objetivo do *Unity* são jogos.

Para executar o projeto feito, pressionou-se CTRL+SHIFT+B no teclado. A opção *Android* foi escolhida dentre as disponíveis e confirmou-se a ação em “*Build And Run*” (Figura 29).

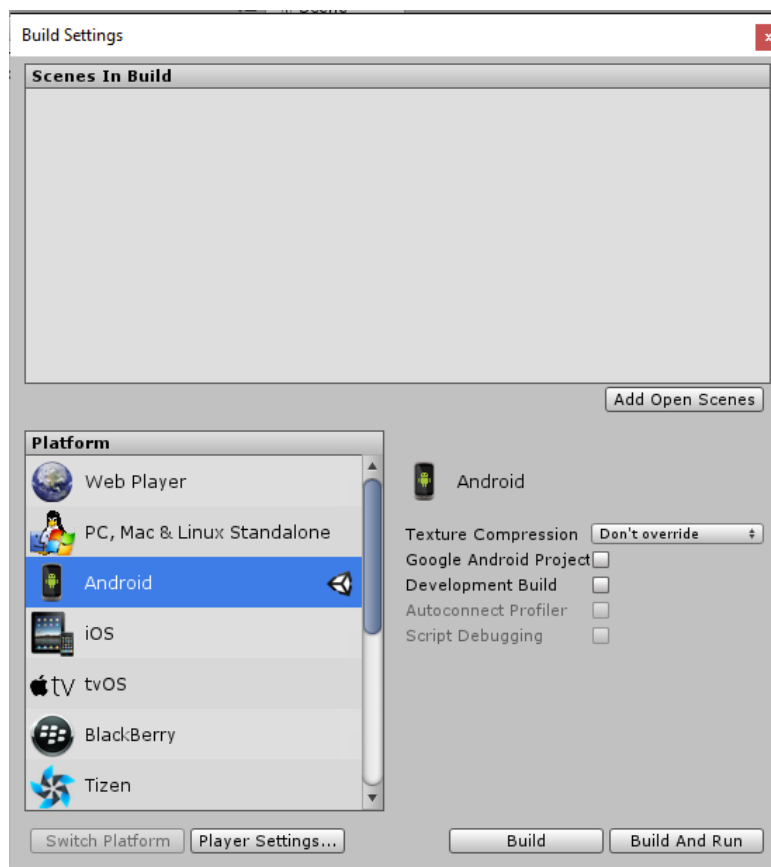


Figura 29 – Selecionar plataforma Unity.
Fonte: Autoria própria.

Automaticamente a aplicação executou no *smartphone* com as seguintes figuras (Figura 30, Figura 31).

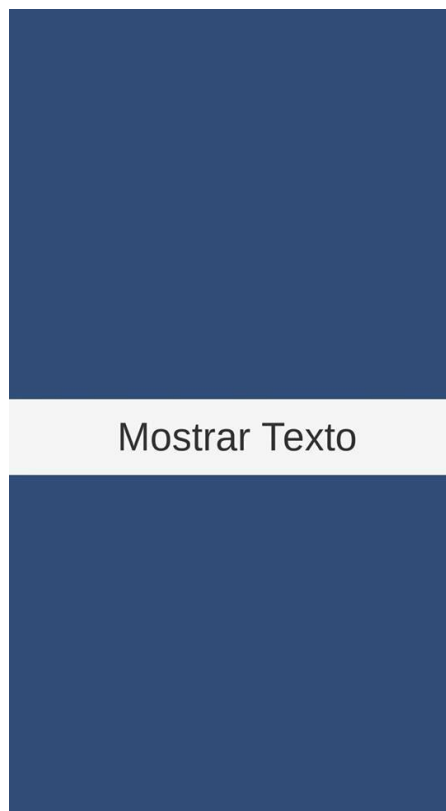


Figura 30 – Tela aplicação Unity.
Fonte: Autoria própria.



Figura 31 – Tela mensagem Unity.
Fonte: Autoria própria.

4.1.4 Projeto Demonstrativo Em Xamarin

Para esta demonstração a IDE *Visual Studio* foi utilizado. Clicando no menu *File -> New -> Project*, uma janela abrirá com opções de que tipo de projeto deseje-se criar. Selecionou-se então a opção *Blank App (Xamarin.Forms Portable)* (Figura 32, Figura 33).

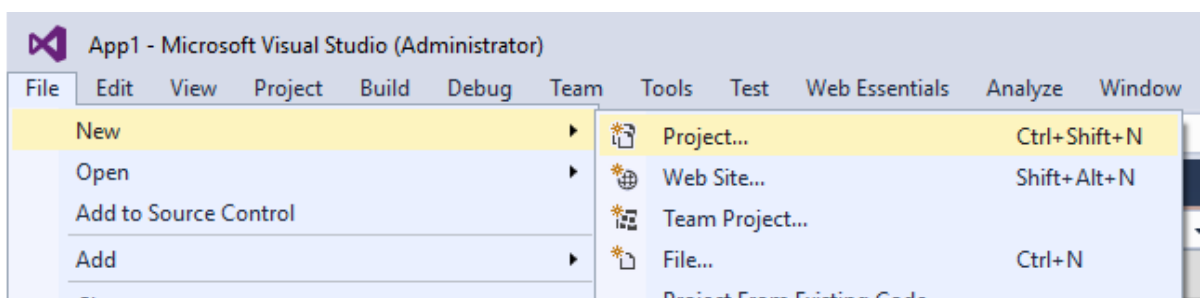


Figura 32 – Criação projeto Xamarin.
Fonte: Autoria própria.

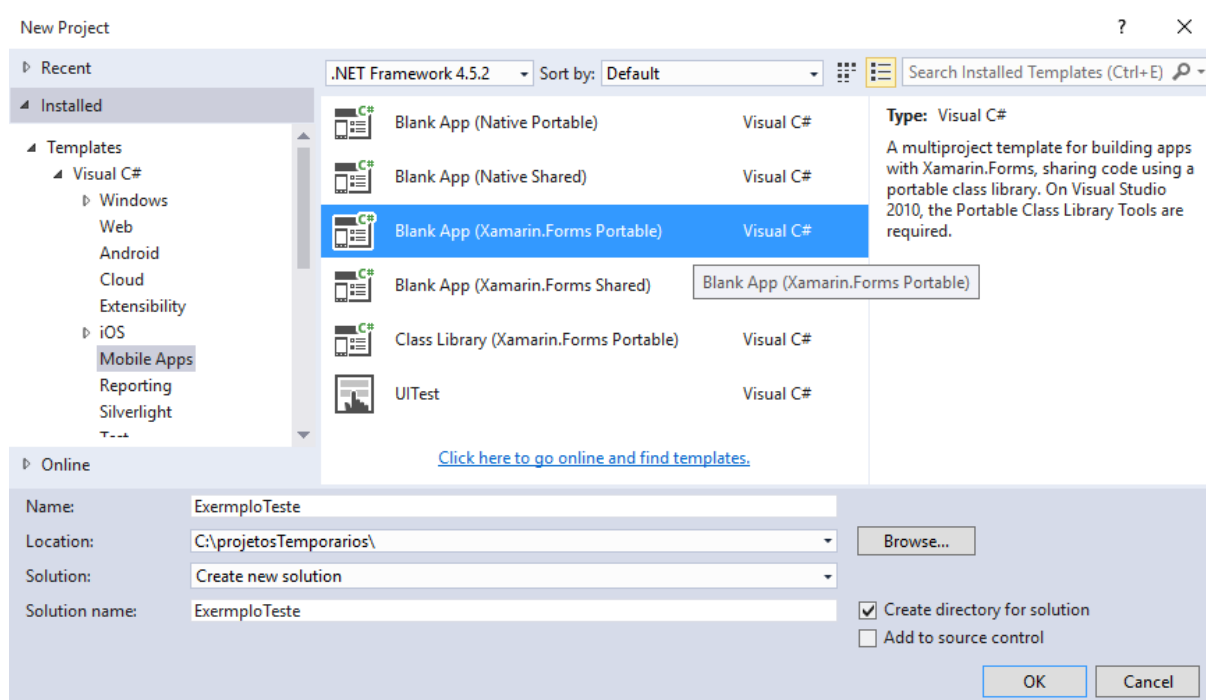


Figura 33 – Tipo de projeto Xamarin.
Fonte: Autoria própria.

Automaticamente quatro projetos são criados, um para cada plataforma suportada, sendo estas: *Android*, *iOS* e *Windows Phone*, e um projeto a parte onde

seu código pode ser compartilhado entre as outras plataformas. O primeiro projeto criado corresponde ao código que pode-se compartilhar entre as plataformas, enquanto código específico para cada uma deve ser escrito em seu respectivo projeto (Figura 34).

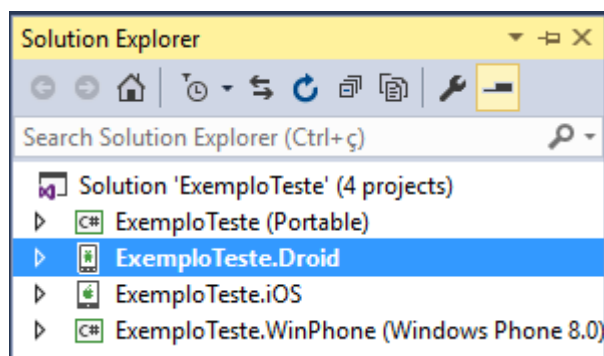


Figura 34 – Estrutura projeto Xamarin.
Fonte: Autoria própria.

Criou-se então um arquivo .XAML no projeto principal de código compartilhado. Para isto as opções *Add -> New Item* foram clicadas.

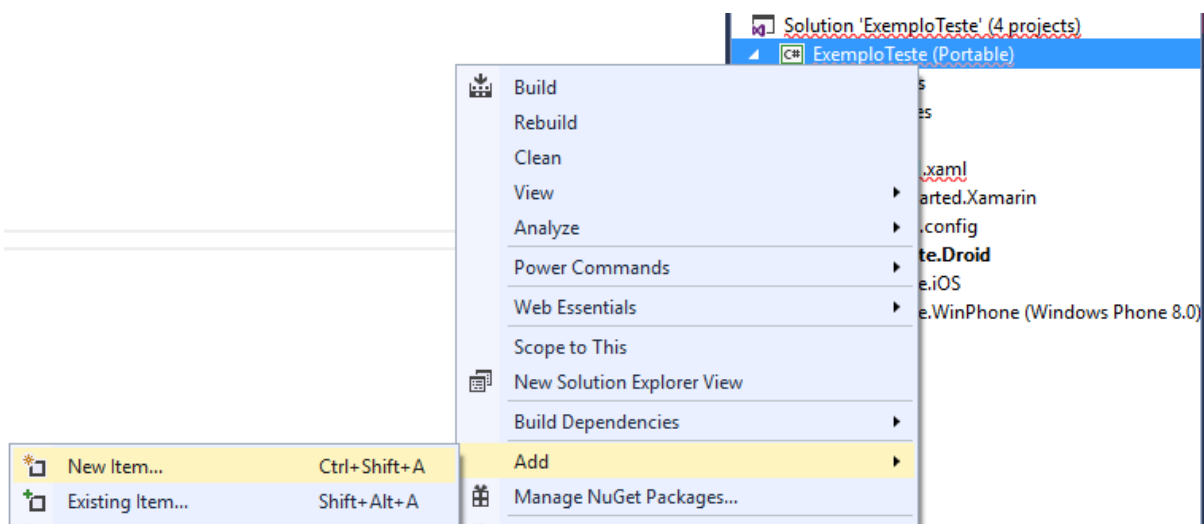


Figura 35 – Adicionar arquivo XAML.
Fonte: Autoria própria.

Na janela que se abriu, a opção *Forms Xaml Page* foi selecionada.

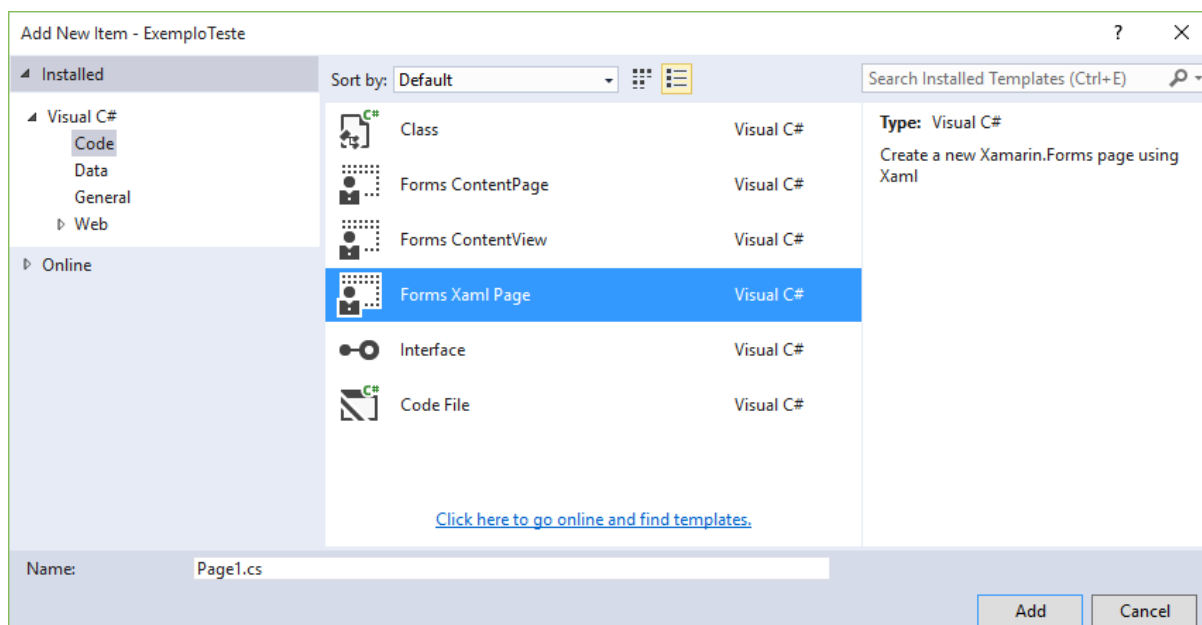


Figura 36 - Selecionar arquivo XAML.
 Fonte: Autoria própria.

Adicionou-se então uma *tag* XML referente a um botão com a referência de uma ação no arquivo criado (**Erro! Fonte de referência não encontrada.**). Ao adicionar a propriedade *Clicked* na *tag*, automaticamente uma opção de criar um método para ela é sugerida, então a tela redirecionou-se para o método em questão.

```
<StackLayout>
  <Button Text="Mostrar Alerta" Clicked="OnClick" />
</StackLayout>
```

Figura 37 – Exemplo código botão Xamarin.
 Fonte: Autoria própria.

```
private void OnClicked(object sender, EventArgs e)
{
    DisplayAlert("Olá UTFPR", "OK");
}
```

Figura 38 – Exemplo código alerta Xamarin.
 Fonte: Autoria própria.

É possível então executar o projeto criado selecionando a plataforma e dispositivo de destino da aplicação, neste caso: Droid e Samsung SM-G920I. O resultado é representado pelas Figura 39 e Figura 40.

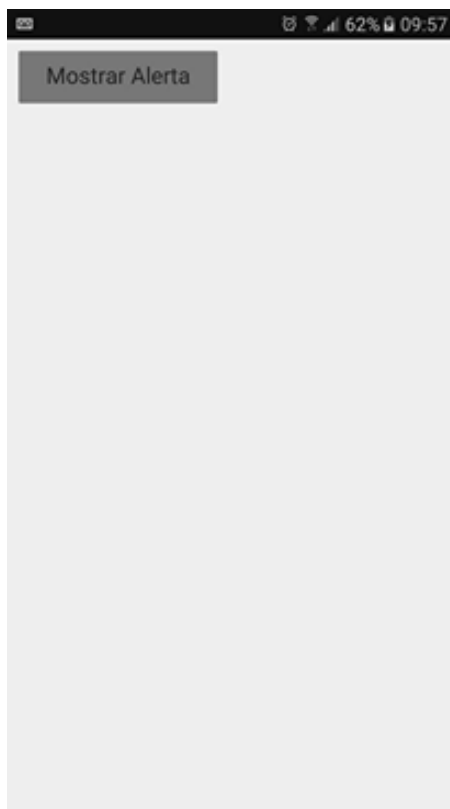


Figura 39 – Tela aplicação Xamarin.
Fonte: Autoria própria



Figura 40 – Tela alerta aplicação Xamarin
Fonte: Autoria própria

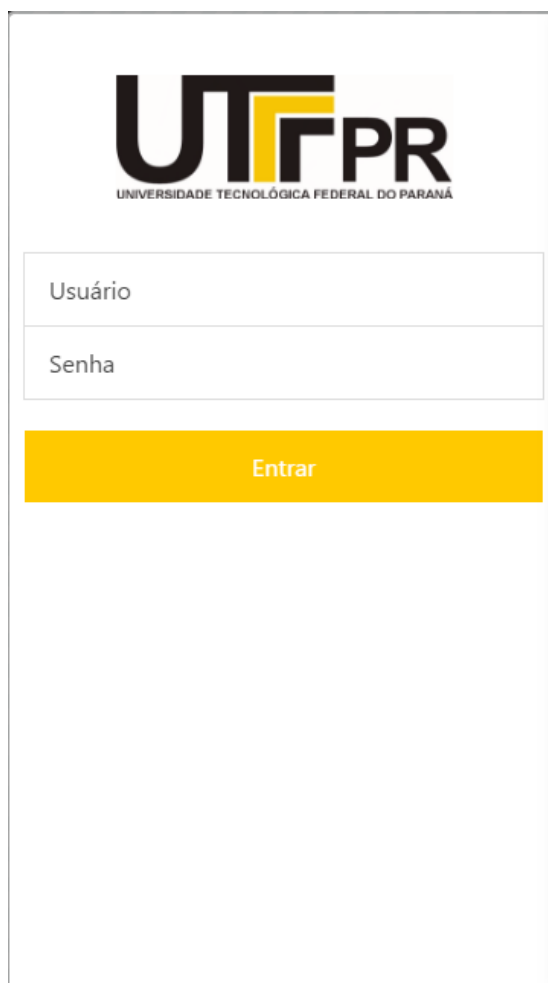
4.1.5 Descrição De *Layout* Das Telas

Desenho das telas do sistema com explicação dos seus respectivos campos.

4.1.5.1 Login de Acesso

4.1.5.1.1 Autenticação de Acesso ao Sistema

Este formulário provê o acesso do usuário ao sistema (Figura 41).



A tela de login do sistema UTFRPR apresenta o logo da Universidade Tecnológica Federal do Paraná (UTFRPR) no topo. Abaixo do logo, há dois campos de entrada para o usuário: 'Usuário' e 'Senha'. Um botão amarelo com o texto 'Entrar' está posicionado abaixo dos campos de entrada.

Figura 41 – Tela de login.
Fonte: Autoria própria.


Usuário: *Login* de acesso do usuário. É utilizado para identificar o usuário.

Senha: Permite ao usuário informar sua senha. Necessária para evitar a entrada de pessoas não autorizadas no sistema.

Entrar: Permite ao usuário entrar no sistema, se for autorizado.

4.1.5.1.2 Mensagem – Usuário ou Senha não informado

Esta mensagem é mostrada quando o usuário não informa o seu *login* ou senha de acesso. Para que o acesso seja permitido o usuário deve estar previamente cadastrado no sistema (Figura 42/Figura 43).



O formulário de login é composto por dois campos de entrada de texto empilhados verticalmente. O campo superior contém o rótulo 'Usuário' e o campo inferior contém o rótulo 'Senha'. Abaixo dos campos, há dois botões de ação: um botão amarelo com o texto 'Entrar' e um botão vermelho com o texto 'Informe o Login e a Senha'.

Figura 42 – Mensagem 1 login.

Fonte: Autoria própria.

4.1.5.1.3 Mensagem – Usuário ou Senha inválidos

Esta mensagem é mostrada quando o usuário não informa o seu *login* ou senha corretamente. Para que o acesso seja permitido o usuário deve preencher os dois campos corretamente (Figura 43).

Usuário	teste
Senha	...

Entrar

Usuário ou senha inválidos

Figura 43 – Mensagem 2 login.
Fonte: Autoria própria.

4.1.5.2 Home da aplicação

Este é um exemplo de ambiente de trabalho do usuário do sistema de controle de viagem (Figura 44). Esta página pode variar de usuário para usuário dependendo do tipo de usuário que o mesmo é. Sendo estes: comum ou administrativo.

Se o usuário ativo for do tipo comum, ele poderá visualizar e terá o poder de finalizar na página inicial apenas as viagens que estão ativas que ele mesmo criou. Apenas o usuário administrador poderá visualizar todas as viagens ativas no momento juntamente com o poder de finalizá-las.

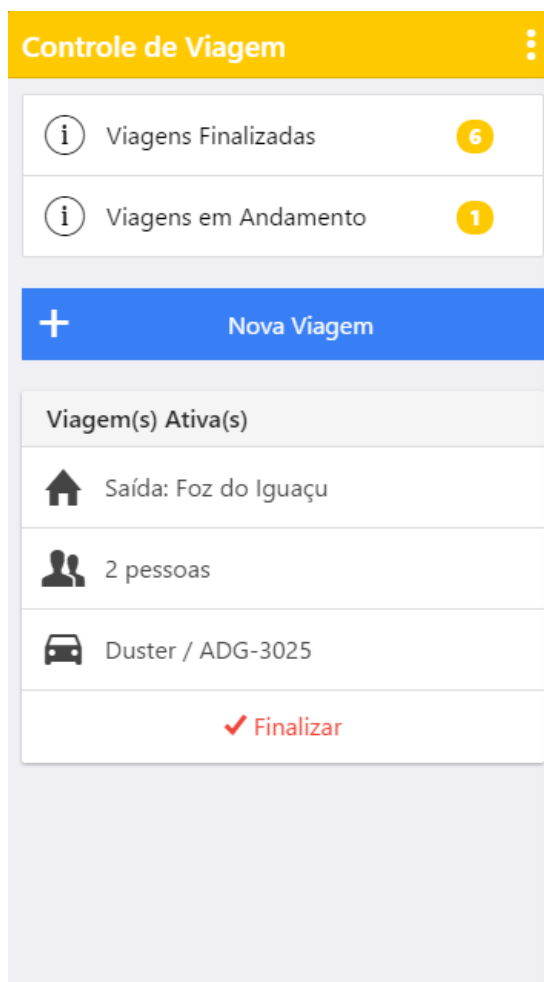


Figura 44 – Tela inicial.
Fonte: Autoria própria.

4.1.5.3 Configurações

Esta tela tem a função de mostrar o usuário que está ativo no sistema e executar a função sair (Figura 45).

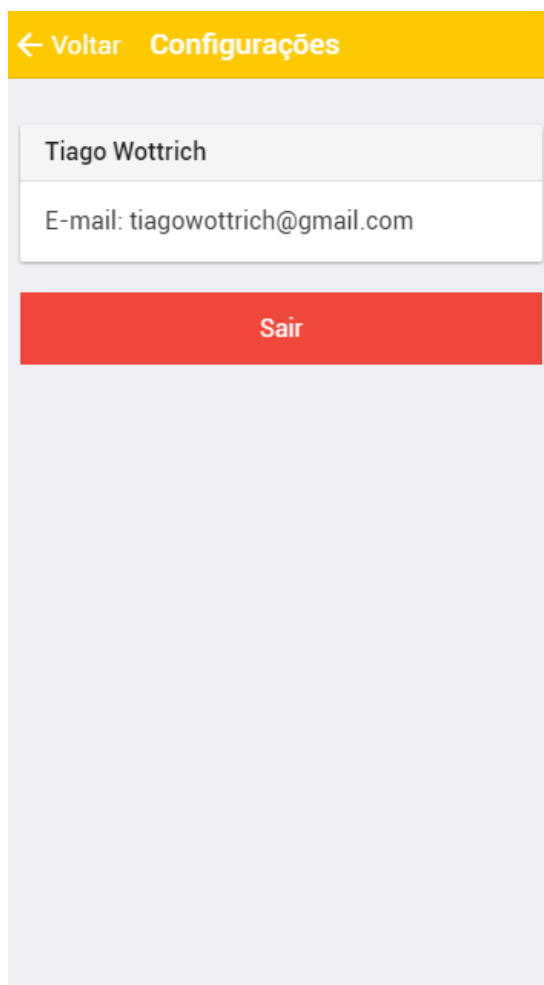


Figura 45 – Tela de configurações.
Fonte: Autoria própria.

4.1.5.4 Pessoas

4.1.5.4.1 Tela Inicial

Esta tela tem a função de mostrar os cinco últimos registros de pessoas cadastrados no sistema, possibilitando também consultar pessoas baseado em seu: Nome, CPF, SIAPE e RA (Figura 46).

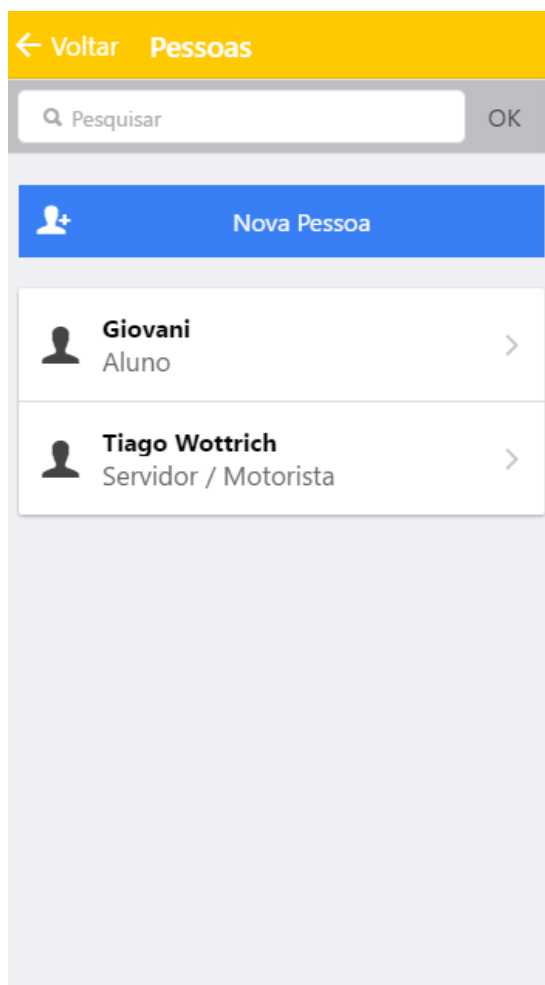


Figura 46 – Tela de pessoas.
Fonte: Autoria própria.

4.1.5.4.1.1 Mensagem – Pessoa não encontrada

Esta mensagem é mostrada quando o usuário informa alguma informação na busca que não corresponde a nenhum dado de nenhuma pessoa registrada no sistema (Figura 47).

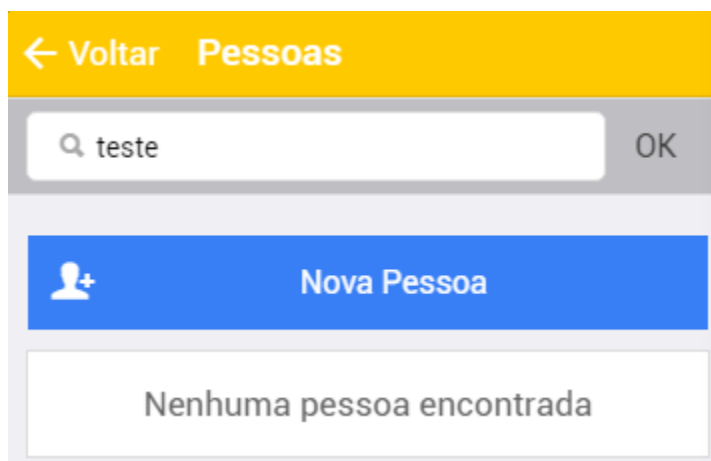


Figura 47 – Consulta de pessoas.
Fonte: Autoria própria.

4.1.5.4.2 Cadastro de Pessoa

Este formulário tem a função de cadastrar pessoas. Se o tipo da pessoa for aluno ou servidor, informações diferentes precisarão ser preenchidas (Figura 48, Figura 49).



← Voltar Nova Pessoa

Nome

CPF

Email

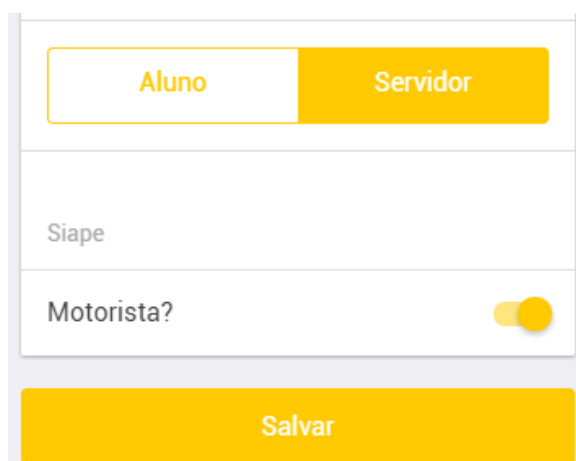
Telefone

Aluno Servidor

RA

Salvar

Figura 48 – Cadastro de pessoa.
Fonte: Autoria própria.



Aluno Servidor

Siape

Motorista?

Salvar

Figura 49 – Cadastro de pessoa parte 2.
Fonte: Autoria própria.

Nome: Nome completo da pessoa.

CPF: Número do CPF (Cadastro de Pessoa Física) da pessoa.

E-mail: Endereço eletrônico a pessoa.

Telefone: Número do telefone para contato com a pessoa.

RA: Número do RA (Registro Acadêmico) do aluno na instituição de ensino.

SIAPÉ: Número identificador do servidor na instituição de ensino.

Motorista?: Caso selecionado o servidor também será considerado como um motorista pelo sistema.

4.1.5.4.2.1 Validações – Campos não informados

Caso no momento de salvar o registro alguma informação obrigatoriamente requerida esteja faltando, automaticamente a aplicação cancela o salvamento e destaca em vermelho no formulário quais são as informações faltantes (Figura 50, Figura 51).

Mesmo que por algum motivo a verificação do formulário falhar, todos os campos são novamente verificados na parte do servidor.

Nome
* Campo Requerido

CPF
* Campo Requerido

Email
* Campo Requerido

Telefone

Aluno Servidor

RA
* Campo Requerido

Figura 50 – Validação cadastro de pessoa.
Fonte: Autoria própria.

Aluno Servidor

Siape
* Campo Requerido

Motorista?

Figura 51 - Validação cadastro de pessoa parte 2.
Fonte: Autoria própria.

4.1.5.4.2.2 Mensagem – CPF inválido

Esta tela de mensagem é mostrada quando o usuário não informa um CPF válido (Figura 52).

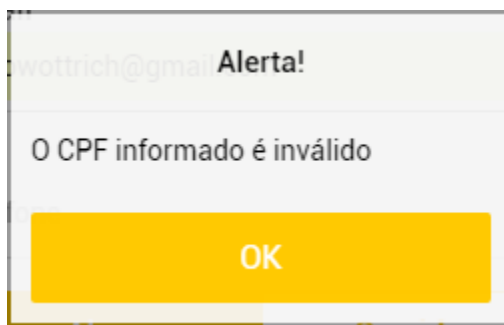


Figura 52 – Mensagem CPF inválido.
Fonte: Autoria própria.

4.1.5.4.2.3 Mensagem – E-mail inválido

Esta tela de mensagem é mostrada quando o usuário não informa um e-mail válido (Figura 53).

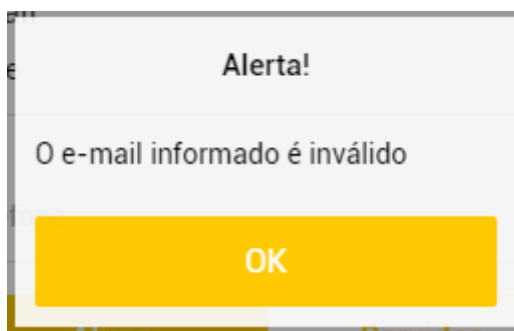


Figura 53 – Mensagem e-mail inválido.
Fonte: Autoria própria.

4.1.5.4.2.4 Mensagem – E-mail já cadastrado

Esta tela de mensagem é mostrada quando o usuário informa um e-mail já registrado para outra pessoa (Figura 54).

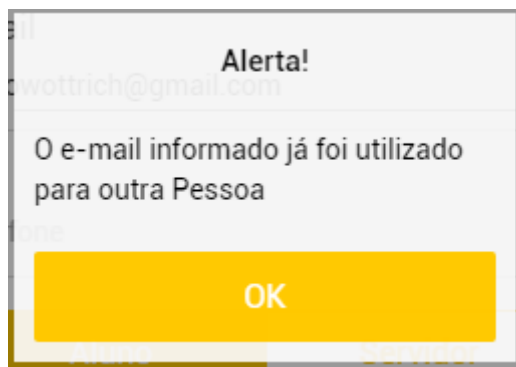


Figura 54 – Mensagem e-mail já cadastrado.
Fonte: Autoria própria.

4.1.5.5 Veículos

4.1.5.5.1 Tela Inicial

Esta tela tem a função de mostrar os cinco últimos registros de veículos cadastrados no sistema, possibilitando também consultar pessoas baseado em seu: Modelo e Placa (Figura 55).

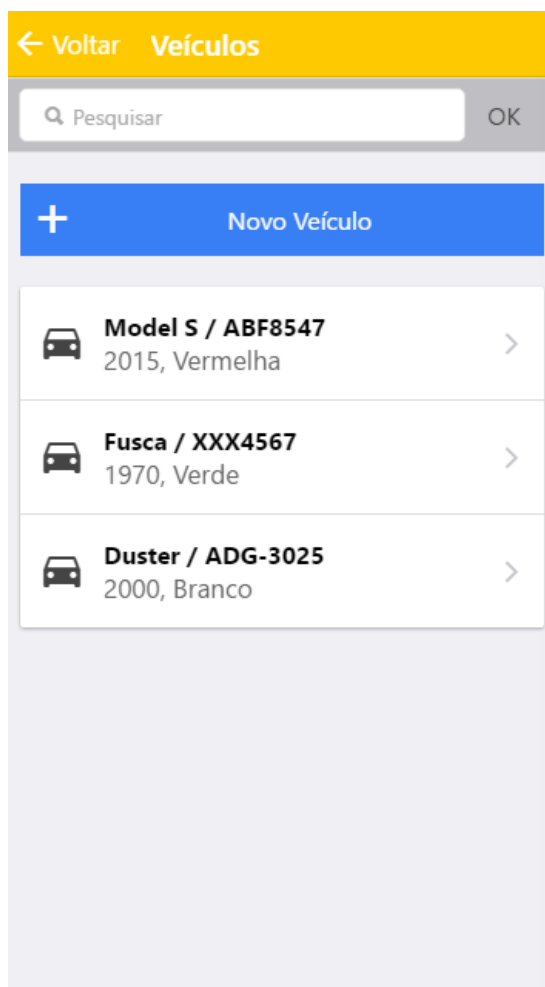
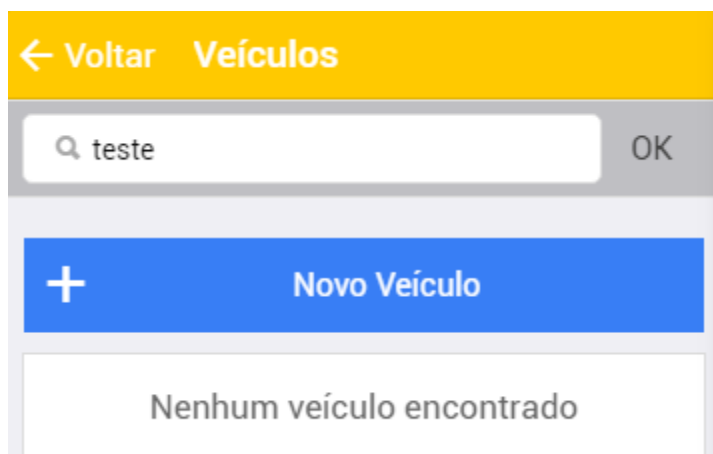


Figura 55 – Tela de veículos.
Fonte: Autoria própria.

4.1.5.5.1.1 Mensagem – Veículo não encontrado

Esta mensagem é mostrada quando o usuário informa alguma informação na busca que não corresponde a nenhum dado de nenhum veículo registrado no sistema (Figura 56).



The screenshot shows a mobile application interface for vehicle management. At the top, there is a yellow header bar with a back arrow and the text 'Voltar Veículos'. Below the header is a search bar containing the text 'teste' and an 'OK' button. Underneath the search bar is a blue button with a white plus sign and the text 'Novo Veículo'. At the bottom of the screen, there is a white box with the text 'Nenhum veículo encontrado'.

Figura 56 – Consulta de veículos.
Fonte: Autoria própria.

4.1.5.5.2 Cadastro de Veículo

Este formulário tem a função de cadastrar veículos (Figura 57).



The screenshot shows a mobile application interface for registering a new vehicle. At the top, there is a yellow header bar with a back arrow and the text 'Voltar Novo Veículo'. Below the header is a form with several input fields: 'Marca', 'Modelo', 'Ano', 'Cor', 'Placa', and 'Quantidade Pessoas' (with the value '0' displayed). Below the 'Quantidade Pessoas' field are two radio buttons labeled 'Ativo' and 'Inativo'. At the bottom of the screen, there is a yellow button labeled 'Salvar'.

Figura 57 – Cadastro de veículo.

Fonte: Autoria própria.

Marca: Marca do veículo.

Modelo: Modelo do veículo.

Ano: Ano do veículo.

Cor: Cor do veículo.

Placa: Número da placa do veículo.

Quantidade de Pessoas: Quantidade máxima de pessoas suportada pelo veículo.

Ativo/Inativo: Situação na qual o veículo se encontra.

4.1.5.5.2.1 Validações – Campos não informados

Caso no momento de salvar o registro alguma informação obrigatoriamente requerida esteja faltando, automaticamente a aplicação cancela o salvamento e destaca em vermelho no formulário quais são as informações faltantes (Figura 58).

Mesmo que por algum motivo a verificação do formulário falhar, todos os campos são novamente verificados na parte do servidor.

Marca * Campo Requerido
Modelo * Campo Requerido
Ano * Campo Requerido
Cor * Campo Requerido
Placa * Campo Requerido
Quantidade Pessoas 0
<input type="radio"/> Ativo <input type="radio"/> Inativo

Figura 58 – Validação cadastro de veículo.
Fonte: Autoria própria.

4.1.5.5.2 Mensagem – Placa já registrada

Esta tela de mensagem é mostrada quando o usuário informa uma placa já cadastrada para outro veículo (Figura 59).

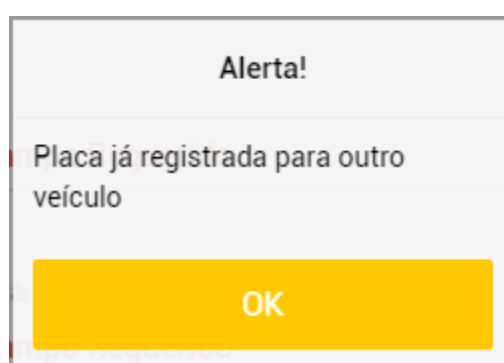


Figura 59 – Mensagem placa já registrada.

Fonte: Aatoria própria.

4.1.5.6 Viagens

4.1.5.6.1 Tela Inicial

Esta tela tem a função de mostrar os cinco últimos registros de viagens cadastrados no sistema (Figura 60). Se o usuário ativo for do tipo comum, ele poderá visualizar apenas as viagens registradas por ele, já o usuário do tipo administrador terá acesso a todas as viagens cadastradas no sistema.

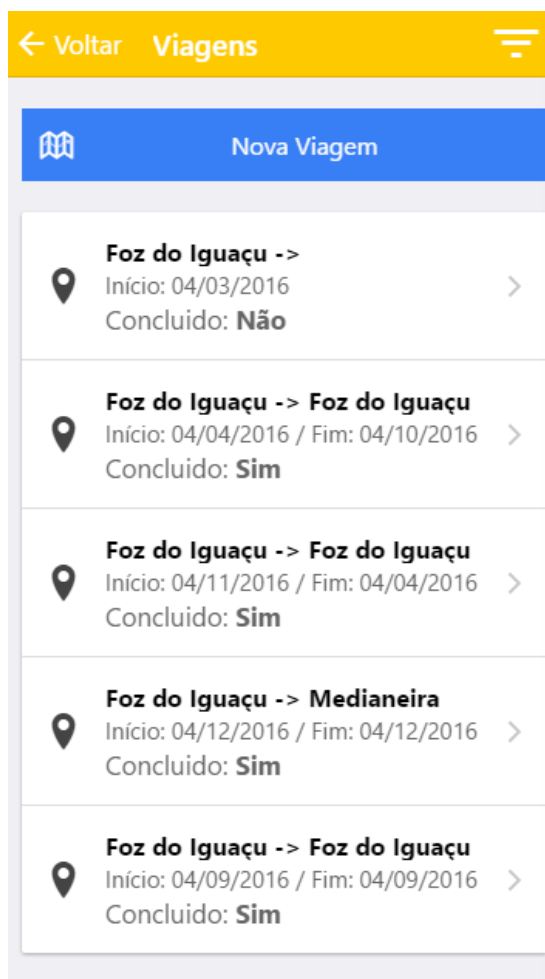
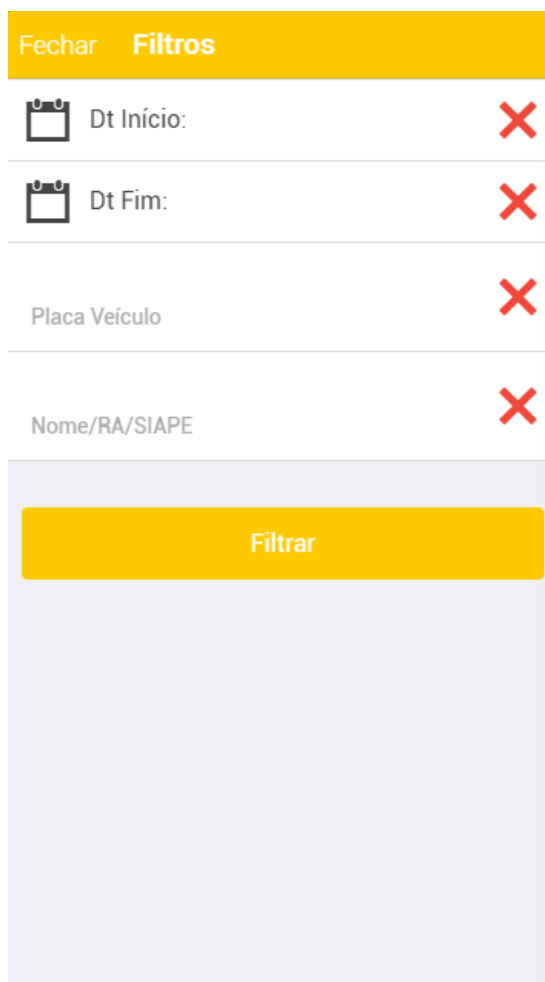


Figura 60 – Tela de viagens.
Fonte: Aatoria própria.

4.1.5.6.1.1 Filtros – Pesquisa de Viagem

Tela com a função de filtrar viagens (Figura 61).



A tela de filtros de viagem apresenta um cabeçalho amarelo com os botões "Fechar" e "Filtros". Abaixo, há quatro campos de filtro, cada um com um ícone de calendário e um botão de exclusão (X) vermelho à direita:

- Dt Início:
- Dt Fim:
- Placa Veículo
- Nome/RA/SIAPE

Na base da tela, há um botão amarelo com o texto "Filtrar".

Figura 61 – Tela de filtros de viagem.
Fonte: Autoria própria.

Dt Início: Data inicial da viagem.

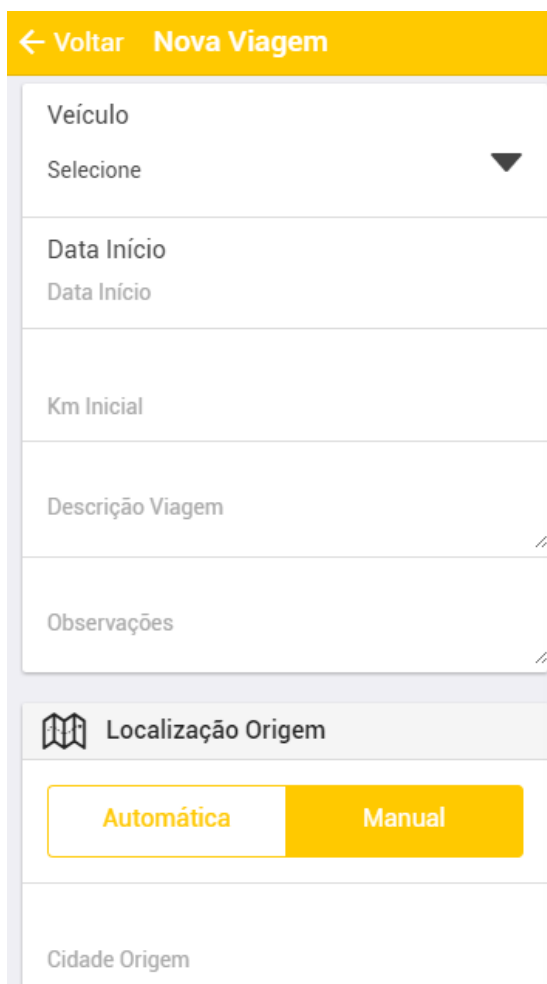
Dt Fim: Data final da viagem.

Placa Veículo: Número da placa do veículo usado na viagem.

Nome/RA/SIAPE: Campo com múltipla capacidade de ser usado para pesquisar tanto o nome, RA ou SIAPE de uma pessoa adicionada à viagem.

4.1.5.6.2 Cadastro de Viagem

Este formulário tem a função de cadastrar viagens (Figura 62, Figura 63).



A tela de cadastro de viagem apresenta um cabeçalho amarelo com o texto "← Voltar Nova Viagem". O formulário contém os seguintes campos:

- Veículo:** Um campo de seleção com o texto "Selecione" e um ícone de seta para baixo.
- Data Início:** Um campo de data com o texto "Data Início".
- Km Inicial:** Um campo de texto.
- Descrição Viagem:** Um campo de texto com uma barra de rolagem.
- Observações:** Um campo de texto com uma barra de rolagem.

Abaixo dos campos, há uma seção intitulada "Localização Origem" com um ícone de mapa. Nesta seção, há dois botões: "Automática" (destacado em amarelo) e "Manual". Abaixo dos botões, há um campo de texto rotulado "Cidade Origem".

Figura 62 – Tela cadastro de viagem.
Fonte: Autoria própria.



A interface de usuário para o cadastro de uma viagem, exibindo três campos de entrada empilhados: 'Endereço', 'Número' e 'Bairro'. Abaixo dos campos, há uma barra de ação com um ícone de pessoas, o texto 'Pessoas' e um ícone de adição (+). Na base da tela, um botão amarelo com o texto 'Salvar' está visível.

Figura 63 – Tela cadastro de viagem parte 2.

Fonte: Autoria própria.

Veículo: Veículo a ser adicionado na viagem previamente cadastrado no formulário de veículos.

Data Início: Data inicial da viagem.

Km Inicial: Quilometragem inicial do veículo.

Descrição Viagem: Descrição da viagem.

Observações: Observações sobre a viagem.

Cidade Origem: Nome da cidade de origem da viagem.

Endereço: Logradouro de origem da viagem.

Número: Número predial de origem da viagem.

Bairro: Bairro da cidade de origem da viagem.

4.1.5.6.2.1 Validações – Campos não informados

Caso no momento de salvar o registro alguma informação obrigatoriamente requerida esteja faltando, automaticamente a aplicação cancela o salvamento e destaca em vermelho no formulário quais são as informações faltantes (Figura 64, Figura 65).

Mesmo que por algum motivo a verificação do formulário falhar, todos os campos e verificações são novamente realizados na parte do servidor.

Veículo Selecione o Veículo
Data Início Data Início * Campo Requerido
Km Inicial * Campo Requerido
Descrição Viagem
Observações

Figura 64 – Validação de viagem.
Fonte: Autoria própria.

 Localização Origem
<input type="radio"/> Automática <input checked="" type="radio"/> Manual
Cidade Origem * Campo Requerido
Endereço * Campo Requerido
Número * Campo Requerido
Bairro * Campo Requerido

Figura 65 – Validação de viagem parte 2.
Fonte: Autoria própria.

4.1.5.6.2.2 Mensagem – Veículo não incluído

Esta tela de mensagem é mostrada quando o usuário não informa um veículo para a viagem (Figura 66).

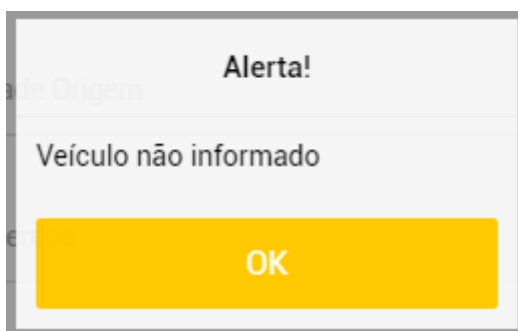


Figura 66 – Mensagem veículo não informado.
Fonte: Autoria própria.

4.1.5.6.2.3 Mensagem – Pessoa não incluída

Esta tela de mensagem é mostrada quando o usuário não informa um veículo para a viagem (Figura 67).

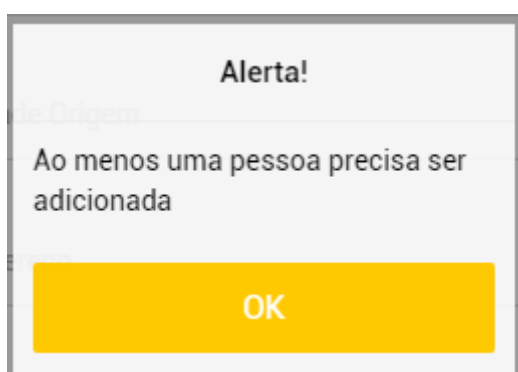


Figura 67 – Mensagem pessoa não incluída.
Fonte: Autoria própria.

4.1.5.6.2.4 Mensagem – Pessoa não é motorista

Esta tela de mensagem é mostrada quando o usuário não informa uma pessoa que seja do tipo motorista para a viagem (Figura 68).

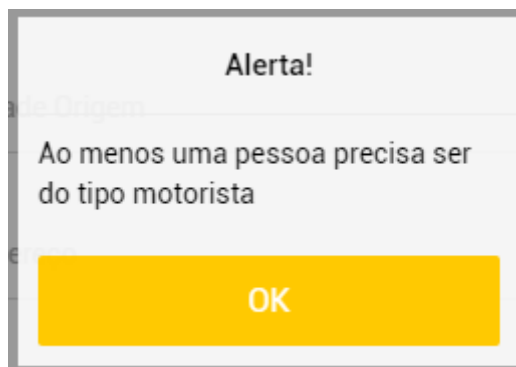


Figura 68 – Mensagem pessoa não é motorista.
Fonte: Autoria própria.

4.1.5.6.2.5 Mensagem – Número de pessoas excedido

Esta tela de mensagem é mostrada quando o usuário adiciona mais pessoas que o suportado pelo veículo para a viagem (Figura 69).

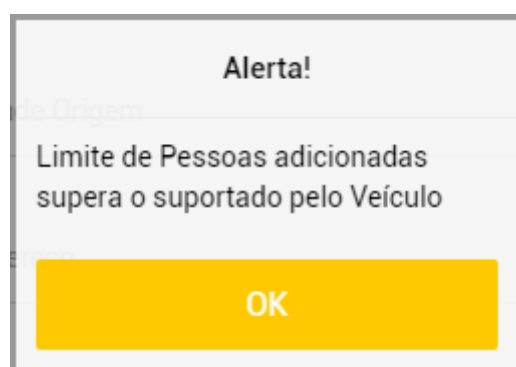


Figura 69 – Mensagem limite de pessoas excedido.
Fonte: Autoria própria.

4.1.5.6.2.6 Mensagem – GPS (*Global Positioning System*) inativo

Esta tela de mensagem é mostrada quando o usuário tenta capturar sua posição automaticamente pelo GPS, porém o mesmo se encontra desativado (Figura 70).

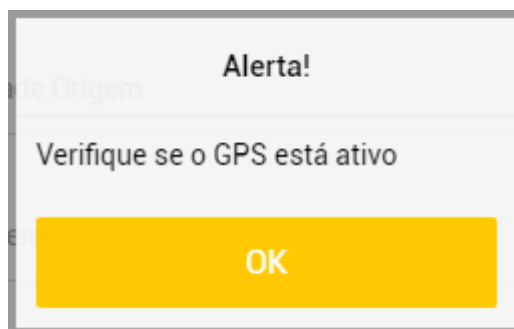


Figura 70 – Mensagem GPS inativo.
Fonte: Autoria própria.

4.1.5.6.3 Finalização viagem

Este formulário tem a função de finalizar viagens cadastradas que estão em andamento (Figura 71, Figura 72).

Fechar Finalizar Viagem

Data Fim
Data Fim

Km Final

Abastecimentos

Descrição Viagem

Observações

Localização Destino

Automática Manual

Cidade Destino

Figura 71 – Tela finalização viagem.
Fonte: Autoria própria.

Endereço

Número

Bairro

Finalizar

Figura 72 - Tela finalização viagem parte 2.
Fonte: Autoria própria.

Data Fim: Data final da viagem.

Km Final: Quilometragem final do veículo.

Abastecimentos: Número de abastecimentos necessários durante a viagem.

Descrição Viagem: Descrição da viagem.

Observações: Observações sobre a viagem.

Cidade Destino: Nome da cidade de destino da viagem.

Endereço: Logradouro de destino da viagem.

Número: Número predial de destino da viagem.

Bairro: Bairro da cidade de destino da viagem.

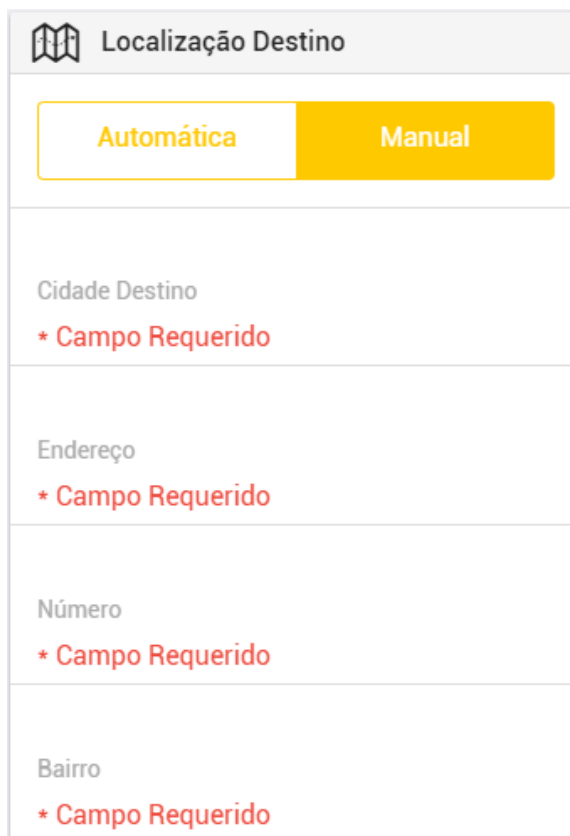
4.1.5.6.3.1 Validações – Campos não informados

Caso no momento de salvar o registro alguma informação obrigatoriamente requerida esteja faltando, automaticamente a aplicação cancela o salvamento e destaca em vermelho no formulário quais são as informações faltantes (Figura 73, Figura 74).

Mesmo que por algum motivo a verificação do formulário falhar, todos os campos e verificações são novamente realizados na parte do servidor.

Data Fim Data Fim * Campo Requerido
Km Final * Campo Requerido
Abastecimentos * Campo Requerido
Descrição Viagem
Observações

Figura 73 – Validação finalização viagem.
Fonte: Autoria própria.



The screenshot shows a form titled "Localização Destino" with a map icon. It features two tabs: "Automática" (highlighted) and "Manual". Below the tabs are four input fields, each with a red asterisk and the text "* Campo Requerido" indicating a validation error:

- Cidade Destino
- Endereço
- Número
- Bairro

Figura 74 - Validação finalização viagem parte 2.
Fonte: Autoria própria.

4.1.5.6.3.2 Mensagem – Data final menor que a inicial

Esta tela de mensagem é mostrada quando o usuário informa uma data final para a viagem menor que a data inicial (Figura 75).

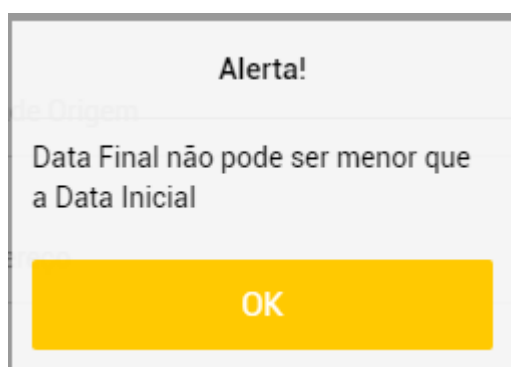


Figura 75 – Mensagem data final menor que inicial.
Fonte: Autoria própria.

4.1.5.6.3.3 Mensagem – Quilometragem final menor que a inicial

Esta tela de mensagem é mostrada quando o usuário informa uma quilometragem final para a viagem menor que a quilometragem inicial (Figura 76).

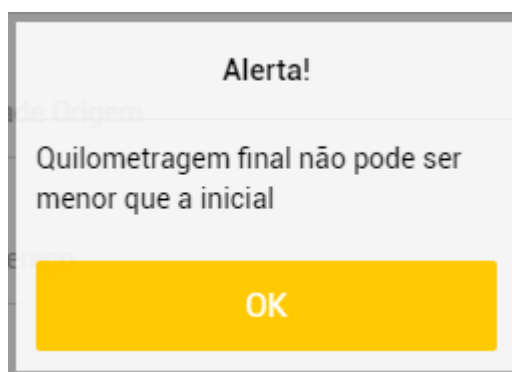


Figura 76 – Mensagem quilometragem final menor que inicial.
Fonte: Autoria própria.

4.1.5.6.4 Visualização da viagem

Este formulário tem a função de recuperar e mostrar a viagem selecionada, expondo todos os seus dados juntamente com seu estado atual, sendo estes: Concluído ou Em Andamento (Figura 77, Figura 78, Figura 79).

Se a viagem selecionada estiver finalizada, um botão com a opção de visualizar o percurso da viagem aparecerá (Figura 80).

Fechar Detalhes Viagem	
Veículo	Duster / ADG-3025
Data Início	04/04/2016
Data Fim	04/10/2016
Km Inicial	2000
Km Final	2555
Km Percorrido	555
Abastecimentos	3
Status	Concluído

Figura 77 – Tela visualização viagem.
Fonte: Autoria própria.


Cadastro	09/04/2016 16:51
Encerramento	09/04/2016 16:52
 Localização Origem	
Cidade Origem	Foz do Iguaçu
Endereço	Avenida Paraná
Número	836
Bairro	Região Central

Figura 78 - Tela visualização viagem parte 2.
Fonte: Autoria própria.




 Localização Destino
Cidade Destino Foz do Iguaçu
Endereço roque pires de barros
Número 2402
Bairro são roque
 Pessoas
 Tiago Wottrich Servidor (Motorista)
Ver Mapa

Figura 79 - Tela visualização viagem parte 3.
Fonte: Autoria própria.

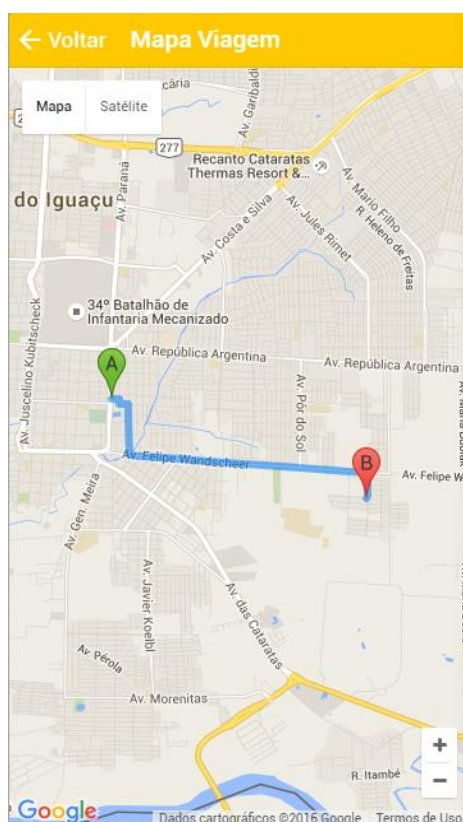


Figura 80 – Tela mapa viagem.
Fonte: Autoria própria.

5 CONSIDERAÇÕES FINAIS

5.1 CONCLUSÕES

Analisando as ferramentas e opções para o desenvolvimento de plataforma cruzada para aplicações móveis, pôde-se verificar que dependendo de qual e que maneira optou-se por desenvolver, o resultado não deixa a desejar perante aplicações 100% nativas. Durante os testes realizados o resultado causou muita surpresa principalmente pela facilidade, economia e produtividade que os meios alternativos propiciam.

Tendo em mente a situação e o objetivo das demonstrações realizadas, o desenvolvimento com Delphi 10 foi o que mais se destacou pela facilidade e rapidez propiciada na execução da demonstração e nenhum tipo de dificuldade foi encontrada. Quanto ao Apache Cordova nenhuma dificuldade foi encontrada também, talvez a parte mais complicada tenha sido lembrar como realizar um evento de clique sem auxílio de bibliotecas como *jQuery*. Utilizar o Xamarin mostrou-se um pouco mais complicado que as demonstrações anteriores, pois apesar de grande parte do código ser gerado automaticamente, foi necessário estudar como criar um botão e atribuir um evento a ele, e mesmo assim não foi tão simples pois existem diversas maneiras de fazer isso, como por exemplo usando um arquivo XML ou não. A demonstração com Unity foi a mais complicada de todas e levou mais de duas horas para ser concluída. Diferente das outras não foi possível encontrar uma maneira simples de exibir uma janela de alerta como nos outros casos, seria necessário criar uma janela e atribuir eventos e funções a ela, por exemplo. Mesmo o ato de atribuir um evento a um botão se mostrou uma tarefa complexa comparada com as demonstrações anteriores, mas no fim foi possível realizar o teste com sucesso.

Quanto à questão de qual meio de desenvolvimento escolher, realmente depende dos requisitos requeridos pelo sistema, tempo e orçamento disponível.

No protótipo proposto em questão, o desenvolvimento com *Apache Cordova* foi escolhido por não ser requisitado muitos itens que dependem do *hardware* do smartphone, sendo usado basicamente requisições *HTTP* para o servidor,

delegando para ele as regras de negócio, deixando para a aplicação em si apenas o trabalho de mostrar as informações. É possível ainda ir mais além do que o protótipo desenvolvido foi, no sentido de fazer uso de capacidades nativas da plataforma, como tirar fotos ou acessar o GPS. Porém deve-se ficar atento quanto a como está sendo desenvolvida a aplicação, pois aplicativos híbridos tendem a ser mais lentos que seus respectivos nativos. Este é um exemplo de limitação entre aplicações híbridas e nativas.

A construção deste trabalho proporcionou uma realização de cunho profissional relevante, visto que é tendência cada vez mais o mercado focar em *smartphones*, e para isso devemos nos preparar. Quem mais souber e estiver preparado terá maior chance de se sobressair perante os outros.

5.2 TRABALHOS FUTUROS/CONTINUAÇÃO DO TRABALHO

Sugere-se uma pesquisa mais elaborada quanto a outros meios disponíveis de desenvolvimento móvel e em como se destacam perante o desenvolvimento nativo.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ADOBE. **Adobe Announces Agreement to Acquire Nitobi, Creator of PhoneGap.** 2011. Disponível em: <<http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html>>. Acessado em Maio/2016.

AMADEO R. **Google's iron grip on Android: Controlling open source by any means necessary.** 2013. Disponível em: <<http://arstechnica.com/gadgets/2013/10/googles-iron-grip-on-android-controlling-open-source-by-any-means-necessary/>>. Acessado em Maio/2016.

AMAZONAS M. **How does Unity scripting work under the hood?.** 2014. Disponível em: <<https://sometimesicodewordpress.com/2014/12/22/how-does-unity-work-under-the-hood/>>. Acessado em Maio/2016.

ANJARWALLA T. **Inventor of cell phone: We knew someday everybody would have one.** 2009. Disponível em: <<http://edition.cnn.com/2010/TECH/mobile/07/09/cooper.cell.phone.inventor/>>. Acessado em Maio/2016.

APPLE. **iPhone App Store Downloads Top 10 Million in First Weekend.** 2008. Disponível em: <<http://www.apple.com/pr/library/2008/07/14iPhone-App-Store-Downloads-Top-10-Million-in-First-Weekend.html>>. Acessado em Maio/2016.

APPEL R. **Modern apps: Mobile Web Sites vs. Native Apps vs. Hybrid Apps.** 2014. Disponível em: <<https://msdn.microsoft.com/en-us/magazine/dn818502.aspx>>. Acessado em Maio/2016

AT&T. **1946: First Mobile Telephone Call.** N/I. Disponível em: <<http://www.corp.att.com/atllabs/reputation/timeline/46mobile.html>>. Acessado em Maio/2016.

BBC. **World's first 'smartphone' celebrates 20 years.** 2014. Disponível em: <<http://www.bbc.com/news/technology-28802053>>. Acessado em Maio/2016.

BINSTOCK A. **.NET Alternative In Transition.** 2011. Disponível em: <<http://www.informationweek.com/architecture/net-alternative-in-transition/d/d-id/1098050?>>. Acessado em Maio/2016.

BRISTOWE J. **What is a hybrid mobile app?** 2015. Disponível em: <<http://developer.telerik.com/featured/what-is-a-hybrid-mobile-app/>>. Acessado em Maio/2016.

BUCHANAN M. **Windows Phone 7 Series: Everything is different now.** 2010. Disponível em: <<http://gizmodo.com/5471805/windows-phone-7-series-everything-is-different-now>>. Acessado em Maio/2016.

CALLAHAM J. **The new Google Store is your one-stop web shop for Chromebooks, Nexus devices and more.** 2015. Disponível em: <<http://www.androidcentral.com/new-google-store-your-one-stop-web-shop-chromebooks-nexus-devices-and-more>>. Acessado em Maio/2016.

CASTELLS M. **A sociedade em rede.** Tradução Roneide V. Majer. 8ª ed. São Paulo: Paz e Terra, 2005. Acessado em Maio/2016.

CHAFFEY D. **Mobile Marketing Statistics Compilation.** 2016. Disponível em: <<http://www.smartinsights.com/mobile-marketing/mobile-marketing-analytics/mobile-marketing-statistics/>>. Acessado em Maio/2016.

CHARLAND A, LEROUX B. **Mobile Application Development: Web vs. Native.** Communications of the ACM, Vol 54, N° 5. 2011. Acessado em Maio/2016.

CRITEO. **State of Mobile Commerce.** 2015. Disponível em: <<http://www.criteo.com/media/1894/criteo-state-of-mobile-commerce-q1-2015-ppt.pdf>>. Acessado em Maio/2016.

EMBARCADERO. **Delphi Feature Matrix.** 2015. Disponível em: <<http://www.embarcadero.com/products/delphi/Delphi-Feature-Matrix.pdf>>. Acessado em Maio/2016.

EMBARCADERO. **FireUI: Revolutionary Multi-Device Development.** 2015. Disponível em: <<https://www.embarcadero.com/br/products/rad-studio/fireui>>. Acessado em Maio/2016.

FILHO A. **“Tempo é dinheiro”: Usabilidade é Produtividade.** 2012. Disponível em: <<http://www.periodicos.uem.br/ojs/index.php/EspacoAcademico/article/view/18526>>. Acessado em Maio/2016.

GARTNER. **Gartner says worldwide smartphone sales grew 9.7 percent in fourth quarter of 2015.** 2016. Disponível em: <<http://www.gartner.com/newsroom/id/3215217>>. Acessado em Maio/2016.

JORGESEN P. **33 mind shattering email marketing statistics you need to know in 2014.** 2014. Disponível em: <<https://expresspigeon.com/blog/2014/01/06/email-marketing-statistics-2014/>>. Acessado em Maio/2016.

INGRAHAM N. **Apple’s App Store has passed 100 billion app downloads.** 2015. Disponível em: <<http://www.theverge.com/2015/6/8/8739611/apple-wwdc-2015-stats-update>>. Acessado em Maio/2016.

KAHN J. **Google show off new version of Android, announces 1 billion active monthly users.** 2014. Disponível em: <<http://www.techspot.com/news/57228-google-shows-off-new-version-of-android-announces-1-billion-active-monthly-users.html>>. Acessado em Maio/2016.

KORF M., OKSMAN E. **Native, HTML5, or Hybrid: Understanding your mobile application development options.** 2015. Disponível em: <

https://developer.salesforce.com/page/Native,_HTML5,_or_Hybrid:_Understanding_Your_Mobile_Application_Development_Options >. Acessado em Maio/2016.

LOOPER J. **What is a WebView?**. 2015. Disponível em: <<http://developer.telerik.com/featured/what-is-a-webview/>>. Acessado em Maio/2016.

LUNDEN I. **Apple took 89% of Q4 smartphone profit with Android OEMs in a race to the bottom.** 2015. Disponível em: <<http://techcrunch.com/2015/02/26/apple-eating-all-the-profits/>>. Acessado em Maio/2016.

MRC. **Native mobile apps: The wrong choice for business?**. N/I. Disponível em: <<http://www.mrc-productivity.com/research/whitepapers/NativeAppsWrongChoice.pdf>>. Acessado em Maio/2016.

NEWSHUB. **Smartphones now outsell 'dumb' phones.** 2013. Disponível em: <<http://www.newshub.co.nz/technology/smartphones-now-outsell-dumb-phones-2013042912#axzz49DmzsH96>>. Acessado em Maio/2016.

O'MALLAY C. **BellSouth's communicative Simon is a milestone in the evolution of the PDA.** 1994. Disponível em: <<http://web.archive.org/web/19990221174856/byte.com/art/9412/sec11/art3.htm>>. Acessado em Maio/2016.

REISINGER D. **Wordwide smartphone user base hits 1 billion.** 2012. Disponível em: <<http://www.cnet.com/news/worldwide-smartphone-user-base-hits-1-billion/>>. Acessado em Maio/2016.

PÁSCOA R. **O que é uma sociedade em rede?** 2011. Disponível em: <<http://mpel.ruipascoa.net/?p=267>>. Acessado em Maio/2016.

RICKER T. **Jobs: App Store launching with 500 iPhone applications, 25% free.** 2008. Disponível em: <<http://www.engadget.com/2008/07/10/jobs-app-store-launching-with-500-iphone-applications-25-free/>>. Acessado em Maio/2016.

SAGER I. **Before iPhone and Android Came Simon, the First Smartphone.** 2012. Disponível em: <<http://www.bloomberg.com/news/articles/2012-06-29/before-iphone-and-android-came-simon-the-first-smartphone>>. Acessado em Maio/2016.

TAKAHASHI D. **John Riccitiello sets out to identify the engine of growth for Unity Technologies.** 2014. Disponível em: <<http://venturebeat.com/2014/10/23/john-riccitiello-sets-out-to-identify-the-engine-of-growth-for-unity-technologies-interview/>>. Acessado em Maio/2016.

UNITY. **Unity - Multiplataform.** 2015. Disponível em: <<http://unity3d.com/unity/multiplatform/>>. Acessado em Maio/2016.

VICTOR. **Android's Google Play beats App Store with over 1 million apps, now officially largest.** 2013. Disponível em: <http://www.phonearena.com/news/Androids-Google-Play-beats-App-Store-with-over-1-million-apps-now-officially-largest_id45680>. Acessado em Maio/2016.

VOGELSTEIN F. **The day Google had to ‘start over’ on Android.** 2013. Disponível em: <<http://www.theatlantic.com/technology/archive/2013/12/the-day-google-had-to-start-over-on-android/282479/>>. Acessado em Maio/2016.

XAMARIN. **What is Xamarin?.** 2014. Disponível em: < <http://xamarin.com/tour> >. Acessado em Fevereiro/2016.

XAMARIN. **Frequently Asked Questions.** 2016. Disponível em: < <https://xamarin.com/faq> >. Acessado em Maio/2016.

ZIEGLER C. **Microsoft talks Windows Phone 7 series development ahead of GDC: Silverlight, XNA, and no backward compatibility.** 2010. Disponível em: <<http://www.engadget.com/2010/03/04/microsoft-talks-windows-phone-7-series-development-ahead-of-gdc/>>. Acessado em Maio/2016.

APÊNDICES

7 APÊNDICE A – CASOS DE USO

Caso de Uso	Manter Pessoas
Descrição	Este caso de uso é responsável pela manutenção do cadastro de Pessoas. Inicia quando o usuário deseja incluir, visualizar ou alterar uma Pessoa do sistema.
Atores	Usuário
Pré-Condições	O usuário de estar identificado no sistema
Fluxo básico	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Pessoas. - Usuário 2. Sistema exibe formulário para manutenção de pessoas. - Sistema 3. Usuário informa os dados da Pessoa. - Usuário 4. Usuário executa o comando para a manutenção dos dados. - Usuário 5. O sistema valida os dados. [E -1], [E -2], [E -3], [E -4],[E - 5],[E - 6],[E - 7], [E - 8], [E - 9]. - Sistema 6. O sistema emite uma mensagem “Pessoa cadastrada com sucesso!”. - Sistema 	
Fluxos alternativos	
FA1 – Alterar os dados da Pessoa	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Pessoas - Usuário 2. Sistema exibe formulário para manutenção de Pessoas - Sistema 3. O usuário seleciona a Pessoa desejada - Usuário 4. O sistema recupera os dados da Pessoa. [E-10] - Sistema 5. Volta ao passo 3. 	
FA2 –Visualização de Pessoa	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Pessoas- Usuário 2. Sistema exibe formulário para manutenção de Pessoas- Sistema 3. O usuário seleciona a Pessoa desejada. - Usuário 4. O sistema recupera os dados da Pessoa. [E-10] - Sistema 	
Exceções	
<p>[E -1] – O usuário não informou o nome da Pessoa. O sistema emite uma mensagem “Nome da Pessoa não informada”.</p> <p>[E -2] – O usuário não informou o CPF da Pessoa. O sistema emite uma mensagem “CPF da Pessoa não informada”.</p> <p>[E -3] – O usuário não informou um CPF válido para a Pessoa. O sistema emite uma mensagem “O CPF informado é inválido”.</p> <p>[E -4] – O usuário não informou o e-mail da Pessoa. O sistema emite uma mensagem “E-mail da Pessoa não informado”</p> <p>[E -6] – O usuário não informou um e-mail válido para a Pessoa. O sistema emite uma mensagem “O e-mail informado é inválido”.</p> <p>[E -7] – O usuário informou um e-mail cadastrado para outra Pessoa do sistema. O sistema emite uma mensagem “O e-mail informado já foi utilizado para outra Pessoa”.</p> <p>[E -8] – O usuário informou que o tipo da Pessoa é Servidor mas não informou o número do SIAPE. O sistema emite uma mensagem “O SIAPE não foi</p>	

informado”.	
[E -9] – O usuário informou que o tipo da Pessoa é Aluno mas não informou o número do RA. O sistema emite uma mensagem “O RA não foi informado”.	
[E -10] – O sistema não encontrou a Pessoa. O sistema emite uma mensagem “Não foi possível encontrar o registro solicitado”.	
Pós-Condições	Uma nova Pessoa foi cadastrada.
Observações	

Quadro 1 – Caso de uso manter pessoa.

Fonte: Autoria própria.

Caso de Uso	Manter Veículos
Descrição	Este caso de uso é responsável pela manutenção do cadastro de Veículos. Inicia quando o usuário deseja incluir, visualizar ou alterar um Veículo do sistema.
Atores	Usuário
Pré-Condições	O usuário de estar identificado no sistema
Fluxo básico	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Veículos. - Usuário 2. Sistema exibe formulário para manutenção de Veículos. - Sistema 3. Usuário informa os dados do Veículo. - Usuário 4. Usuário executa o comando para a manutenção dos dados. - Usuário 5. O sistema valida os dados. [E -1], [E -2], [E -3], [E -4],[E - 5],[E - 6],[E – 7], [E – 8].- Sistema 6. O sistema emite uma mensagem “Veículo cadastrado com sucesso!”. - Sistema 	
Fluxos alternativos	
FA1 – Alterar os dados do Veículo	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção do Veículo- Usuário 2. Sistema exibe formulário para manutenção de Veículos - Sistema 3. O usuário seleciona o Veículo desejado - Usuário 4. O sistema recupera os dados do Veículo. [E-10] - Sistema 5. Volta ao passo 3. 	
FA2 - Visualização de Veículo	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Veículos - Usuário 2. Sistema exibe formulário para manutenção de Veículos - Sistema 3. O usuário seleciona o Veículo desejado. - Usuário 4. O sistema recupera os dados do Veículo. [E-10] - Sistema 	
Exceções	
[E -1] – O usuário não informou o nome da marca do Veículo. O sistema emite uma mensagem “Marca do Veículo não informada”.	
[E -2] – O usuário não informou o nome do modelo do Veículo. O sistema emite uma mensagem “Modelo do Veículo não informado”.	
[E -3] – O usuário não informou um ano para o Veículo. O sistema emite uma mensagem “Ano do Veículo não informado”.	
[E -4] – O usuário não informou a placa do Veículo. O sistema emite uma mensagem “Placa do Veículo não informada”.	
[E - 5] - O usuário informou uma placa já cadastrada no sistema. O sistema emite	

<p>uma mensagem “Placa já registrada para outro veículo”.</p> <p>[E -6] - O usuário não informou a cor do Veículo. O sistema emite uma mensagem “Cor do Veículo não informada”.</p> <p>[E -7] – O usuário não informou a quantidade de passageiros do Veículo. O sistema emite uma mensagem “Quantidade de passageiros não informado”.</p> <p>[E -8] – O sistema não encontrou o Veículo. O sistema emite uma mensagem “Não foi possível encontrar o registro solicitado”.</p>	
Pós-Condições	Um novo Veículo foi cadastrado.
Observações	

Quadro 2 – Caso de uso manter veículos.

Fonte: Autoria própria.

Caso de Uso	Manter Viagens
Descrição	Este caso de uso é responsável pela manutenção do cadastro de Viagens. Inicia quando o usuário deseja incluir, visualizar ou finalizar uma Viagem do sistema.
Atores	Usuário
Pré-Condições	O usuário de estar identificado no sistema
Fluxo básico	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Viagens. - Usuário 2. Sistema exibe formulário para manutenção de Viagens. - Sistema 3. Usuário informa os dados da Viagem. - Usuário 4. Usuário executa o comando para a manutenção dos dados. - Usuário 5. O sistema valida os dados. [E -1], [E -2], [E -3], [E -4],[E - 5],[E - 6],[E – 7], [E – 8], [E – 9], [E – 10], [E – 21]. - Sistema 6. O sistema emite uma mensagem “Viagem cadastrada com sucesso!”. - Sistema 	
Fluxos alternativos	
FA1 – Finalizar Viagem	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção da Viagem- Usuário 2. Sistema exibe formulário para manutenção de Viagens - Sistema 3. O usuário seleciona a Viagem desejada - Usuário 4. O sistema recupera os dados da Viagem. [E-10] - Sistema 5. Usuário executa o comando para a manutenção dos dados. - Usuário 6. O sistema valida os dados. [E -12], [E -13], [E -14], [E -15],[E - 16],[E - 17],[E – 18], [E – 19], [E – 20], [E – 21]. - Sistema 7. O sistema emite uma mensagem “Viagem finalizada com sucesso!”. - Sistema 	
FA2 –Visualização da Viagem	
<ol style="list-style-type: none"> 1. O usuário inicia a manutenção de Viagens - Usuário 2. Sistema exibe formulário para manutenção de Viagens - Sistema 3. O usuário seleciona a Viagem desejada. - Usuário 4. O sistema recupera os dados da Viagem. [E-11] - Sistema 	
Exceções	
<p>[E -1] – O usuário não informou um Veículo. O sistema emite uma mensagem “Veículo não informado”.</p> <p>[E -2] – O usuário não informou o nome da cidade de origem da viagem. O</p>	

sistema emite uma mensagem “Cidade de origem não informada”.

[E - 3] – O usuário não informou o nome do logradouro do endereço de origem da viagem. O sistema emite uma mensagem “Nome do logradouro de origem não informado”.

[E - 4] – O usuário não informou o número predial do endereço de origem da viagem. O sistema emite uma mensagem “Número predial do endereço de origem não informado”.

[E - 5] - O usuário não informou o nome do bairro do endereço de origem da Viagem. O sistema emite uma mensagem “Nome do bairro do endereço de origem não informado”.

[E - 6] - O usuário não informou a data inicial da Viagem. O sistema emite uma mensagem “Data Inicial não informada”.

[E - 7] - O usuário não informou a quilometragem inicial do Veículo. O sistema emite uma mensagem “Quilometragem inicial não informada”.

[E - 8] - O usuário não adicionou nenhuma pessoa para a Viagem. O sistema emite uma mensagem “Ao menos uma pessoa precisa ser adicionada”.

[E - 9] - O usuário não adicionou nenhuma pessoa do tipo motorista para a Viagem. O sistema emite uma mensagem “Ao menos uma pessoa precisa ser do tipo motorista”.

[E - 10] - O usuário adicionou mais pessoas para a viagem do que o suportado pelo Veículo. O sistema emite uma mensagem “Limite de Pessoas adicionadas supera o suportado pelo Veículo”.

[E - 11] – O sistema não encontrou a Viagem. O sistema emite uma mensagem “Não foi possível encontrar o registro solicitado”.

[E - 12] - O usuário não informou o nome da cidade de destino da viagem. O sistema emite uma mensagem “Cidade de destino não informada”.

[E - 13] – O usuário não informou o nome do logradouro do endereço de destino da viagem. O sistema emite uma mensagem “Nome do logradouro de destino não informado”.

[E - 14] - O usuário não informou o número predial do endereço de destino da viagem. O sistema emite uma mensagem “Número predial do endereço de destino não informado”.

[E - 15] - O usuário não informou o nome do bairro do endereço de destino da Viagem. O sistema emite uma mensagem “Nome do bairro do endereço de destino não informado”.

[E - 16] - O usuário não informou a data final da Viagem. O sistema emite uma mensagem “Data Final não informada”.

[E - 17] - O usuário informou uma data final de Viagem menor que a data inicial. O sistema emite uma mensagem “Data Final não pode ser menor que a Data Inicial”.

[E - 18] - O usuário não informou a quilometragem final do Veículo. O sistema emite uma mensagem “Quilometragem final não informada”.

[E - 19] – A quilometragem final é menor que a quilometragem inicial. O sistema emite uma mensagem “Quilometragem final não pode ser menor que a inicial”.

[E - 20] – O Usuário não informou o número de abastecimentos da viagem. O sistema emite uma mensagem “Número de abastecimentos não informado”.

[E - 21] – O Usuário tenta capturar sua localização automaticamente pelo GPS (*Global Positioning System*), porém ele se encontra desativado. O sistema emite uma mensagem “Número de abastecimentos não informado”.

Pós-Condições

Uma nova Viagem foi cadastrada.

Observações	Não deve haver duas viagens ativas com o mesmo Veículo. Não deve haver duas viagens ativas com a(s) mesma(s) Pessoa(s).
--------------------	--

Quadro 3 – Caso de uso manter viagens.

Fonte: Autoria própria.