

Carona Não - Solução para Divisão de Gastos com Combustível em Deslocamentos

**Carlos Henrique de Macêdo, Paulo Henrique Rodrigues Abreu,
Janielly Aliny Araújo Silva, Elthon Oliveira**

¹ Universidade Federal de Alagoas - *Campus Arapiraca*
Núcleo de Ciências Exatas (NCEX)
Laboratório Multidisciplinar de Computação / Coletivo EIDI

carlosmacedo025@gmail.com,

{paulo.abreu, janielly.silva, elthon}@arapiraca.ufal.br

Abstract. *It is common among university students, teachers, workers, among others, to share the same car for a trip in common. This article aims to present the application titled Carona Não, which aims to solve a problem common to people who share the fuel value spent on group travel. The application Carona Não aims to ensure that the amount charged to each person participating in the trip is fair, not to generate disagreements or constraints. In addition, it is possible to store information about frequent passengers, and may associate debts or credits with them.*

Resumo. *É comum entre alunos universitários, professores, trabalhadores, entre outros, compartilharem o mesmo carro para uma viagem em comum. Este artigo tem como objetivo apresentar o aplicativo intitulado Carona Não, que pretende resolver um problema comum às pessoas que dividem o valor do combustível gasto em viagens em grupo. O aplicativo Carona Não objetiva garantir que o valor cobrado a cada pessoa participante da viagem seja justo, para não gerar desavenças ou constrangimentos. Além disso, é possível guardar informações sobre passageiros frequentes, podendo associar dívidas ou créditos aos mesmos.*

1. Introdução

Devido ao preço elevado da gasolina, as pessoas estão optando cada vez mais por utilizarem transportes públicos, ou dividir os custos em caronas com amigos. A divisão de custo em caronas pode ser bem vantajosa, por ser mais cômodo e seguro que o transporte público. Porém, existe alguns problemas, tais como ficar cobrando aos passageiros a cada viagem, ou anotado no caderno o quanto os passageiros estão devendo. Dessa forma, seria mais prático cobrar os passageiros por mês.

A situação descrita acima pode ser bastante desconfortável. Fazer o cálculo para encontrar o custo não é difícil, mas se torna massante se for algo corriqueiro, como ir para o trabalho, faculdade, etc. Isso pode fazer com que o motorista não queira fazer cálculos e, por conta disso, decida cobrar valores que julgue apropriados. Nesse contexto, o condutor pode arcar com prejuízo, ou acabar cobrando mais do que deveria, o que em ambos os casos não é justo.

Em meio a essa problemática é que surgiu a ideia de criar o aplicativo *Carona Não*, seu objetivo é facilitar a organização das dívidas resultantes no hábito das *caronas*. Dessa forma, se algum passageiro não tiver dinheiro no ato da viagem, a dívida pode ser paga em um outro momento. Tudo é registrado pelo aplicativo, mas o que torna a aplicação usual de fato é a forma como ela consegue calcular, tentando ser o mais justo possível.

Se uma pessoa parte de um ponto X para um ponto Z, e no trajeto são apanhados 2 amigos, é justo que quem estiver mais próximo do ponto de destino Z, pague menos que a pessoa que saiu do ponto X. Essa é a lógica básica implementada na aplicação em questão. Nessa circunstância, são considerados quantos litros o carro faz por quilômetro, qual a quilometragem total percorrida na viagem, e a partir de qual ponto um determinado passageiro foi apanhado.

2. Materiais e Métodos

As ferramentas utilizadas para o desenvolvimento do aplicativo “*Carona Não*” são:

- AngularJS
- Cordova
- Ionic Framework
- SQLite

A seguir, cada uma dessas ferramentas são detalhadas

2.1. Framework Ionic

O Ionic é um *framework* de código aberto usado para desenvolvimento de aplicativos híbridos [Ionicframework 2018]. A utilização desta ferramenta permite que desenvolvedores a partir de uma aplicação, usando alguns outros frameworks que trabalham em conjunto com o Ionic, consigam desenvolver para várias plataformas. Sem a necessidade de criar uma aplicação nativa para cada variação de dispositivo.

Além disso, o Ionic é um *framework* que contém alguns recursos de componentes de interface que facilitam a construção das telas das aplicações. De modo que essas contêm uma estrutura responsiva¹, assim como nas aplicações *web*, mas com aparência *mobile*. Ademais, o Ionic usa tecnologias como o AngularJS e o Cordova na construção das aplicações.

O AngularJS é um *framework* que permite a criação de páginas em HTML com exibições dinâmicas [Angular 2018]. Este apesar de utilizar o JavaScript, usufrui de um processo chamado de transpilação, que consiste na conversão de uma linguagem em outra. Nesse contexto, é utilizado o TypeScript que ao ser *transpilado* gera um código JavaScript que alimenta a aplicação. A vantagem de adotar o TypeScript é a utilização de conceitos de programação de alto nível de forma mais abstrata. Além da tipagem dos dados, que reduz erros de desenvolvimento e de usuário.

O Cordova “oferece um grupo de APIs que permitem desenvolver uma aplicação com HTML, CSS e JavaScript encapsulada como aplicação móvel nativa, possibilitando assim o acesso às funções nativos dos dispositivos” [NetBeans 2018]. Dessa forma, o Ionic utiliza o AngularJS para codificação das aplicações, em seguida o Cordova a partir do código *transpilado* pelo AngularJS gerar um aplicativo para a plataforma desejada.

¹É a capacidade de uma aplicação se adequar a qualquer formato de tela.

2.2. SQLite

“O SQLite é um mecanismo de banco de dados SQL incorporado”[SQLite 2018], que pode ser utilizado em várias plataformas. O aplicativo *Corona Não* usa desse mecanismo no Ionic através de um *plugin* disponibilizado pelo Cordova, que permite que os dados sejam salvos localmente no dispositivo. Dessa forma o dispositivo funciona sem a necessidade de está conectado a internet, visto que a aplicação deve ser usada em viagens que nem sempre é possível a conectividade com a rede móvel ou fixa.

3. Carona Não

O aplicativo “*Carona Não*” está disponível apenas para *Android*, e tem com principais funcionalidades:

- Cadastrar passageiros
- Calcular o consumo de gasolina por passageiros
- Guardar histórico de dívida ou crédito de passageiros

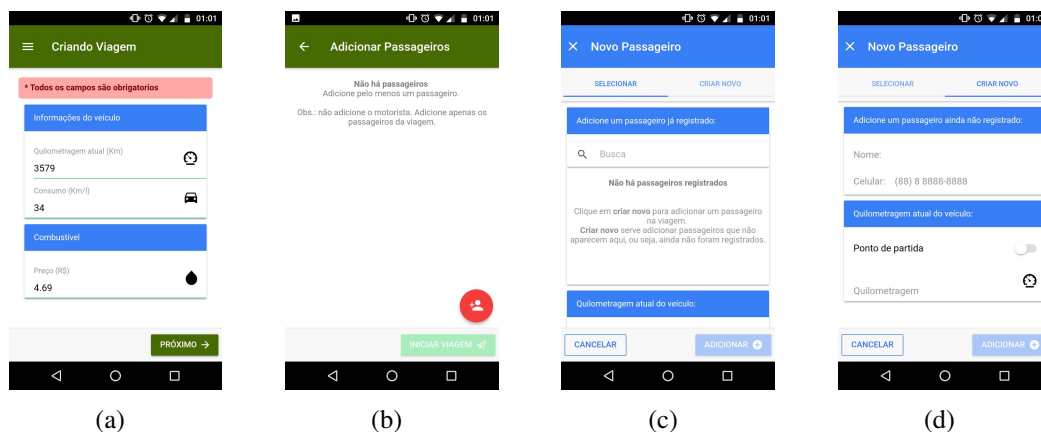


Figura 1. Telas da aplicação

Na Figura 1(a) é ilustrada a página inicial onde existem três campo obrigatórios de entrada de dados para criação de uma viagem, são eles: a quilometragem atual do carro, o consumo do carro por km/l e o preço atual do combustível. A próxima página como mostrado na Figura 1(b) são inseridos os passeiros que participarão da viagem, caso os passeiros não estejam cadastrados ou selecionados, é possível realizar essas ações de forma rápida através no botão vermelho localizando no canto inferior direito (Figura 1(b)), que será direcionado a tela de cadastro e seleção de passageiros como pode ser visto na Figura 1(c).

Para realização do cadastro é necessário o nome, celular e o ponto de partida que o passageiro embarcou na viagem. Esse ponto de partida pode ser a origem da viagem ou qualquer outro ponto durante o trajeto no qual o passageiro tenha sido apanhado (Figura 1(d)). Ao selecionar todos os passageiros é possível iniciar a viagem(Figura 2(a), se durante o percurso houver mais algum passageiro é possível voltar para tela anterior e adiciona-lo a viagem (Figura 2(b)).

Ao finalizar a viagem é necessário pressionar o botão “Calcular” como ilustrado na Figura 2(b) quando pressionado o usuário precisam informar a quilometragem final da

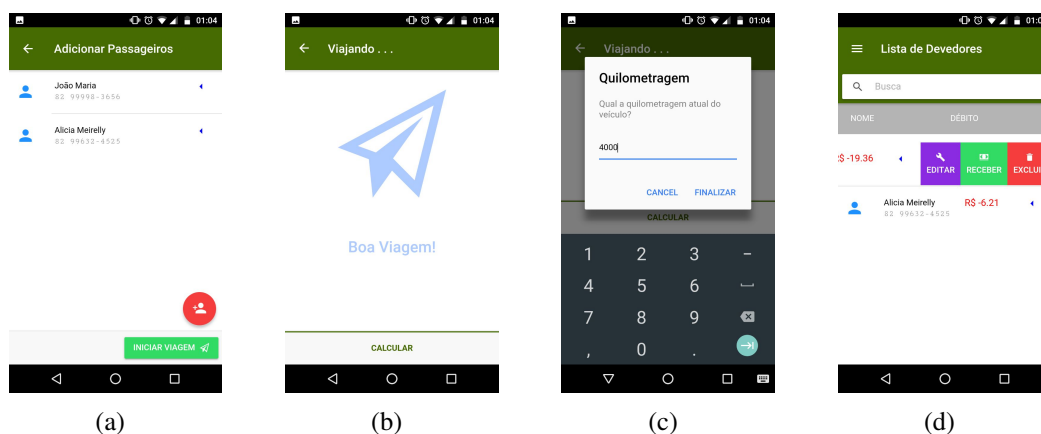


Figura 2. Telas da aplicação

viagem como visto na Figura 2(c). Após todo o processo o aplicativo direciona o usuário para a tela na Figura 2(d) que lista todos os passageiros devedores e suas respectivas dívidas, é possível remover e atualizar os dados dos devedores, ou receber pagamentos.

Uma nova versão está em desenvolvimento para tornar o aplicativo mais completo, utilizando o recurso de GPS, novos modos de divisão de combustível, histórico de custos por viagens, e compartilhamento de dados com os passageiros. Além disso, calcule levando em consideração mais variáveis, quantidade de pedágios, pontos de paradas, etc.

4. Conclusão

A aplicação se encontra publicado na loja Play Store², atualmente tem cerca de 300 *downloads* e todo seu código está disponível no GitHub³, onde qualquer pessoa pode contribuir. Apesar de uma nova versão está em desenvolvimento com mais recursos, o aplicativo pode ser usado sem nenhum problema. Nenhum *bug* foi relatado pelos usuários e até então a aplicação se mantém com a mesma versão desde de sua publicação.

Diante desse cenário, o “Carona Não” é um aplicativo que resolve um problema real, presente na vida de muitas pessoas. De modo, que a aplicação ajuda na gerencia, para facilitar a divisão do custo total de foram justa para que todos envolvidos paguem apenas o que “consumiu”, e de forma indireta manter essa cultura de divisão que resulta em uma economia significativa para cada um dos envolvidos.

Referências

Angular (2018). Angular: What is angular? <https://angular.io/docs>. Acessado em 20/05/2017.

Ionicframework (2018). The top open source framework for building amazing mobile apps. <https://ionicframework.com/framework>. Acessado em 30/05/2018.

²<https://play.google.com/store/apps/details?id=com.coletivoeidi.caronanao>

³<https://github.com/ColetivoEIDI/carona-nao-mobile>

NetBeans (2018). Conceitos básicos sobre criação de uma aplicação cordova. https://netbeans.org/kb/docs/webclient/cordova-gettingstarted_pt_BR.html. Acessado em 01/06/2017.

SQLite (2018). Query language understood by sqlite. <https://www.sqlite.org/lang.html>. Acessado em 13/06/2017.