

# UM PWA PARA CONTROLE DE CUSTOS DE VEÍCULOS UTILIZANDO IONIC

Daniel Granville dos Santos\*  
Anubis Graciela de Moraes Rossetto †

2021

## Resumo

Este trabalho apresenta um progressive web app (PWA) para controle de custos de veículos do usuário. A ideia é proporcionar ao usuário uma maneira ágil de registrar, pelo seu dispositivo móvel, todos os custos que são gerados pelo seu veículo. Além disso, com algumas informações fornecidas inicialmente pelo usuário, o app pode alertar sobre a necessidade de algum tipo de manutenção, por exemplo, troca de óleo e filtros. Para o desenvolvimento optou-se por utilizar o framework Ionic que possibilita a criação de aplicativos híbridos multiplataforma, sendo para este projeto adotada a geração de um PWA que tem como característica a similaridade com um app nativo, porém executado no navegador. Para persistência dos dados foi utilizada a plataforma Cloud Firestore, uma solução de armazenamento remoto.

**Palavras-chave:** custos de veículo. PWA. Ionic.

## Introdução

Os custos para manter um ou mais veículos da família sempre pesam no orçamento, sendo muitas vezes relacionados a manutenções não previstas. Porém, algumas ações de acompanhamento podem melhorar o planejamento do orçamento, ou seja, conhecer os custos que o veículo gera, bem como ter uma previsão de futuros

---

\* <[dani17pf@gmail.com](mailto:dani17pf@gmail.com)>

† <[anubisrossetto@ifsul.edu.br](mailto:anubisrossetto@ifsul.edu.br)>

gastos. Além disso, a não realização de manutenções periódicas do veículo podem gerar mais custos no futuro e ainda aumentar risco de acidentes.

Assim, um planejamento pode envolver o controle de manutenções corretivas e preventivas. A manutenção corretiva refere-se a correção de falhas do veículo, ou seja, significa que peças deixaram de funcionar corretamente e estão impactando o seu funcionamento ou inviabilizando sua utilização, sendo estas imprevisíveis. Já a manutenção preventiva é a análise e substituição das peças do carro dentro de períodos estipulados previamente. Inspeções regulares são importantes pois estão relacionadas com a garantia do veículo e, claro, com o seu bom desempenho e utilização.

Considerando esses aspectos, esse trabalho propôs o desenvolvimento de uma aplicação usando a abordagem PWA para controlar os custos de veículos, com uma interface fácil e intuitiva. O aplicativo possibilita ao usuário cadastrar os seus veículos, gerenciar os seus custos rotineiros e lançar alertas para manutenções preventivas.

Dado que a tecnologia está cada vez mais presente no nosso cotidiano e com recursos inovadores, especialmente no que diz respeito aos dispositivos móveis, podemos ter muitos benefícios usando esses recursos que são acessíveis na palma da nossa mão. Recentemente, há um progresso importante no desenvolvimento de aplicações para dispositivos móveis que permitem o uso de recursos web, denominada Progressive Web Apps (PWA).

De acordo com Luiz (2021) "os Progressive Web Apps (PWA) são desenvolvidos e elaborados com APIs modernas para fornecer recursos aprimorados, confiabilidade e capacidade de instalação, ao mesmo tempo que alcançam qualquer pessoa, seja qual for o lugar, em todos dispositivos com uma única base de código". A aplicação PWA possibilita acesso por meio de uma URL na web, facilitando ao usuário que não precisa passar pelo processo de instalação de um aplicativo.

Com isso, a proposta é fornecer um PWA que permite fazer o registro dos veículos do usuário, seus abastecimentos e manutenções, além de configurar alertas das manutenções preventivas. Com algumas informações fornecidas inicialmente pelo usuário, o app pode alertar sobre a necessidade de algum tipo de manutenção, por exemplo, troca de óleo e filtros. Também, com essas informações o app pode gerar relatórios de custos por período e mostrar alertas sempre que o usuário acessar a aplicação.

Para o desenvolvimento optou-se por utilizar o framework Ionic e para persistência dos dados está sendo utilizado o serviço Cloud Firestore da plataforma Firebase. Essa solução foi a escolhida para garantir que os dados sejam preservados, algo que não seria possível se o armazenamento fosse local e ocorresse algum problema com o dispositivo do usuário.

Esse artigo está organizado da seguinte forma: seção 1 apresenta as características da abordagem de desenvolvimento PWA, bem como as tecnologias usadas para desenvolvimento da aplicação. A Seção 2 apresenta alguns trabalhos relacionados. Na seção 3 estão aspectos sobre a modelagem da aplicação e o detalhamento sobre o seu desenvolvimento. Por fim, são apresentadas as considerações finais.

# 1 Tecnologias para desenvolvimento móvel

A seguir são apresentadas as características da metodologia de desenvolvimento PWA e as tecnologias que foram utilizadas para o seu desenvolvimento: Framework Ionic e Plataforma Firebase.

## 1.1 Progressive Web App (PWA)

Progressive Web App são aplicativos da web, progressivamente aprimorados com tecnologias da web modernas.

Os Progressive Web Apps são apenas ótimos sites que podem se comportar como aplicativos nativos - ou, talvez, os Progressive Web Apps são apenas ótimos aplicativos, alimentados por tecnologias da Web e fornecidos com infraestrutura da Web. [Microsoft \(2021\)](#)

Progressive Web App é uma metodologia de desenvolvimento móvel que busca transformar o dispositivos mobile na Web podendo ser acessados também pelos navegadores de outros dispositivos, por exemplo, um desktop.

Conforme [Russell \(2015\)](#), as características de uma aplicação PWA são relacionadas abaixo.

- **Responsivo**

Possui a capacidade de se adaptar a qualquer tamanho de tela com sua responsividade, tanto para desktop como para celulares e tablets.

- **Progressiva**

Qualquer usuário pode acessar o conteúdo disponível, tanto pelo navegador como pelo dispositivo smartphone;

- **Independente de conectividade**

Consegue trabalhar off-line ou em redes de baixa qualidade;

- **Semelhante a aplicativos**

A navegação e a estrutura são semelhantes a um aplicativo instalável, com o intuito de facilitar visualmente para o usuário;

- **Seguro**

É trafegado utilizando https, sendo assim as informações da aplicação PWA não podem ser modificadas ou adulteradas;

- **Atualizada**

Está sempre atualizado devido a um serviço que roda em plano de fundo;

- **Linkável**

Sendo compartilhado e acessado através de uma URL, facilitando o acesso, sem a necessidade de instalação da aplicação;

- **Instalável**

Permite que os usuários adicionem aplicativos a suas telas principais, através diálogos do navegador, sem a necessidade de acessar uma loja de aplicativos;

- **Reengajamento**

Facilidade com as ferramentas e seus recursos, como as notificações push;

- **Descobrível**

Pode ser identificado através dos manifestos W3C e pelo escopo de registro do service worker, que permitem que os mecanismos de pesquisa os encontrem.

## 1.2 Framework Ionic

De acordo com [Grillo \(2017\)](#), "Ionic é um *framework* para desenvolvimento de aplicações para dispositivos móveis que visa o desenvolvimento de apps híbridas e de rápido e fácil desenvolvimento".

Nos últimos anos observou-se uma grande demanda por recursos tecnológicos que permitissem maior praticidade e agilidade na realização de tarefas cotidianas, seja em casa, no trabalho, nos momentos de lazer ou até mesmo no trânsito. Tais exigências, trouxeram eletrônicos portáteis e outras tecnologias agregadas, tais como GPS e WiFi, que permitiram aos usuários inúmeras possibilidades de utilização e ocupando cada vez mais espaço na vida das pessoas.

Como isso, fica perceptível o crescente nicho de mercado para a área de desenvolvimento de aplicativos, sendo uma preocupação constante que estes possam rodar em diferentes plataformas, tenham uma interface simples e sejam fáceis de usar.

Conforme [Viebrantz e Campos \(2015\)](#), com a utilização do *framework* Ionic é possível a criação de aplicativos híbridos permitindo o desenvolvimento multiplataforma, sendo possível instalar a aplicação em dispositivos Android, iOS e Windows Phone. Também consegue-se utilizar dos recursos nativos, fornecidos pelo *hardware* do dispositivo, como a utilização da câmera, vibração, leitor de QR code, geolocalização entre muitos outros recursos. O *framework* Ionic permite o desenvolvimento com uso das tecnologias *frontend*, como HTML, CSS e Javascript.

Para a construção da aplicação são utilizadas diversas linguagens que se comunicam. De acordo com [Bispo et al. \(2012\)](#), "a linguagem HTML, desenvolvida em 1989, é uma linguagem baseada em marcas, onde palavras-chave indicam o início e o fim de cada elemento do hipertexto". Ainda, de acordo com os autores, usando a linguagem de marcação HTML, é possível construir o corpo da aplicação, em que os elementos são marcados para identificar quais informações a página exibe.

Cascade Style Sheets (CSS), traduzido como folhas de estilo em cascata, refere-se a um mecanismo utilizado para controlar a aparência de páginas Web ([BISPO et al., 2012](#)). Dessa forma, o CSS é uma linguagem de estilo, com a qual é possível realizar a estilização da aparência de uma página Web, ou seja, por meio da utilização do CSS é possível dar forma aos elementos HTML. Para isso, com o

uso do CSS seleciona-se um elemento do HTML para estilizá-lo e personaliza sua aparência, como mudar a fonte de um texto ou alterar a cor do fundo da aplicação.

Segundo o [Flanagan \(2004\)](#), Javascript é uma linguagem programada para adicionar comportamentos dinâmicos, para isso ela permite identificar eventos e a partir disso gerar ações, como, por exemplo, identificar o clique em um botão, para a partir desse evento realizar uma ação de apresentar ao usuário uma mensagem de alerta.

De acordo com a documentação [ionicframework \(2019\)](#) "Os aplicativos Ionic são feitos de blocos de construção de alto nível chamados Components, que permitem construir rapidamente a interface do usuário para o seu aplicativo. O Ionic vem com uma série de componentes, incluindo cartões, listas e guias.". A estrutura iônica é baseada na biblioteca angular e concentra-se na criação rápida de aplicativos da Web por meio de uma variedade de componentes predefinidos para o projeto e interações da interface do usuário.

Existem diversos *frameworks* que facilitam e auxiliam na tarefa de criação de um aplicativo, entre os mais utilizados atualmente está o *framework* Ionic, de acordo com [SILVA \(2018\)](#).

### 1.3 Plataforma Firebase

A Firebase é a plataforma Google que oferece ferramentas para desenvolvedores de seus projetos e, portanto, podendo assim desenvolver ou criar aplicações de alta qualidade. ([FIREBASE, 2021d](#)).

Essa plataforma da Google é utilizada para facilitar a implementação, onde o desenvolvedor adiciona um projeto e inclui serviços que desejam usar conforme sua necessidade.

A plataforma Firebase contém uma documentação dos recursos disponíveis de cada um de seus serviços, facilitando para o desenvolvedor implementar suas funcionalidades. Cada serviço conta com uma explicação de como proceder. Porém nem todos os serviços são gratuitos e alguns deles contêm um limite de utilização ([FIREBASE, 2021e](#)).

Ele oferece diversos recursos, entre eles alguns que foram usados na implementação desse projeto são abordados a seguir.

- Cloud Firestore

O Cloud Firestore é um banco de dados flexível e escalonável para desenvolvimento focado em dispositivos móveis ([FIREBASE, 2021a](#)). Responsável por armazenar e sincronizar os dados NoSQL do aplicativo desenvolvido, o Firestore tem uma estrutura diferente da Realtime Database: os dados são divididos em coleções(*collections*) e documentos(*documents*) ao invés de nós (nodes), e como o Firebase Realtime Database, ele mantém seus dados em sincronia em aplicativos cliente por meio de listeners em tempo real.

- Authentication

O Firebase Authentication fornece serviços de back-end, SDKs fáceis de usar e bibliotecas de IU prontas para autenticar usuários no aplicativo (FIREBASE, 2021b). Ele facilita o processo de autenticação através de contas do Google, Facebook, Twitter, Github entre outras contas.

- Firebase Hosting

O Firebase Hosting oferece hospedagem rápida e segura para apps da Web, conteúdo estático e dinâmico e microsserviços (FIREBASE, 2021c). É um serviço que facilita para os desenvolvedores criarem e hospedarem PWAs, podendo ser acessados por uma URL de qualquer lugar pelo navegador.

## 2 Trabalhos Relacionados

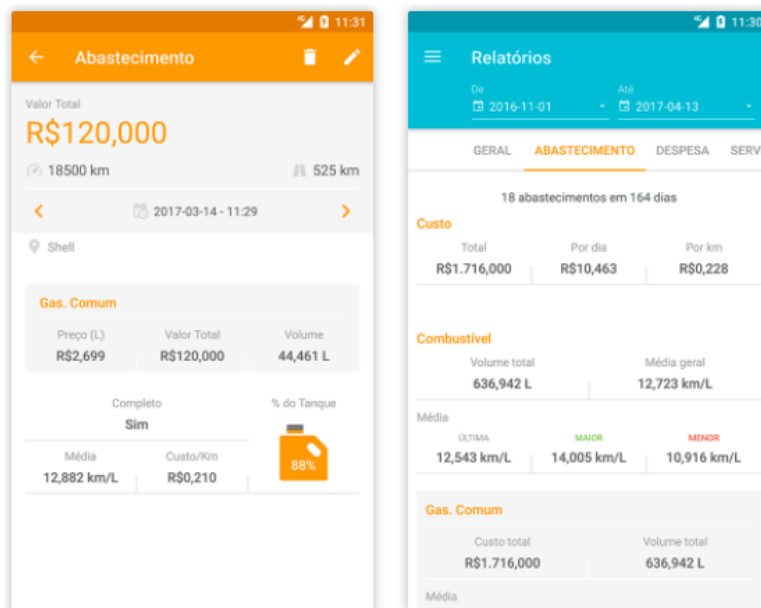
Nesta seção são apresentadas algumas aplicações que tem funcionalidades semelhantes a aplicação desenvolvida.

### 2.1 Drivvo

O Drivvo é a ferramenta direcionada a conservação e a gestão de custos de veículos drivvo (2021). É uma ferramenta que funciona como um gerenciador dos custos, bem como para anotar despesas com abastecimento, manutenção e impostos (como IPVA, licenciamento ou multas).

É possível analisar as datas de fornecimento, com a quantidade e custo de combustível para saber em detalhes o consumo feito, podendo ver cada item e suas quantidades nos relatórios. A Figura 1 mostra uma tela do aplicativo.

Figura 1 – Drivvo



Fonte: drivvo (2021)

## 2.2 Calcular Combustível

Conforme Santos (2021), o aplicativo realiza o cálculo de quanto é o consumo médio do veículo, quanto irá gastar para percorrer uma determinada distância, quantos litros irá abastecer com um determinado valor e a diferença se a opção for álcool. A Figura 2 mostra uma tela do aplicativo.

Para aqueles que procuram uma alternativa mais simples, o cálculo do combustível é uma boa opção. Depois de informar a quantidade de quilômetros, consumo de combustível e quantidade paga por isso, o aplicativo retorna automaticamente várias informações.

Figura 2 – Calcular Combustível



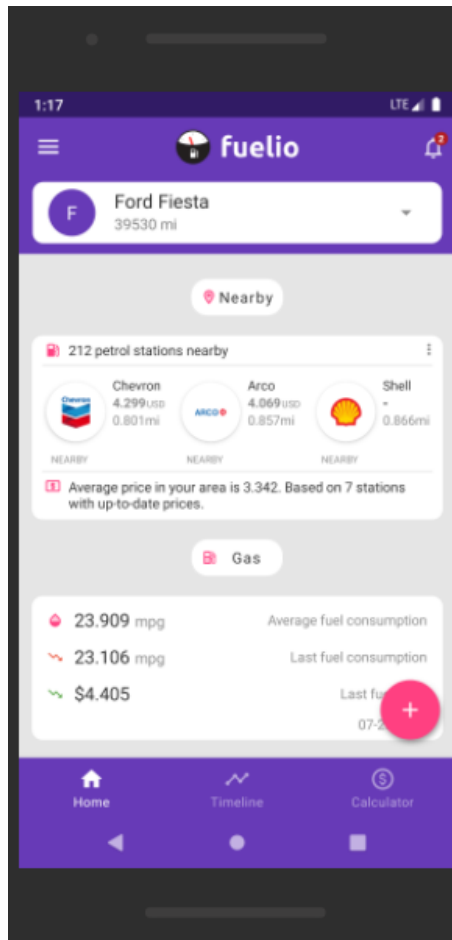
Fonte: Santos (2021)

## 2.3 Fuelio

De acordo com Fuelio (2021), o aplicativo Fuelio é simples e fácil de usar para rastrear a quilometragem, consumo de combustível e custos com combustível. A Figura 3 mostra uma tela do aplicativo.

O aplicativo propõem registrar a quilometragem, consumo e despesas de combustível. Sendo assim, o usuário pode acompanhar as despesas, serviços, suprimentos, consumo de combustível, quilometragem, preços de combustível.

Figura 3 – Aplicativo Fuelio



Fonte: Fuelio (2021)

### 3 Aplicação proposta

Nesta seção são apresentadas a modelagem da aplicação do sistema e o detalhamento sobre o seu desenvolvimento.

#### 3.1 Modelagem da aplicação

O aplicação proposta busca controlar os custos dos veículos do usuário. O usuário pode cadastrar vários veículos e os controles existentes são: registrar abastecimentos e registrar manutenções, sempre selecionando o veículo para a operação. Quanto as manutenções, são todos os custos que o usuário tem com o veículo, desde um reparo de pneu, quanto troca de pneus, troca de óleo, alinhamento e geometria, troca de filtros, entre outros. Como já mencionado, as manutenções podem ser corretivas (por exemplo, reparo de um pneu que furou) ou preventivas (troca de óleo). Os dois tipos de manutenções podem ser registrados no aplicativo. De outro lado, é importante que o usuário tenha um controle da periodicidade das manutenções preventivas que seguem a especificação do fabricante. Assim, a especificação pode



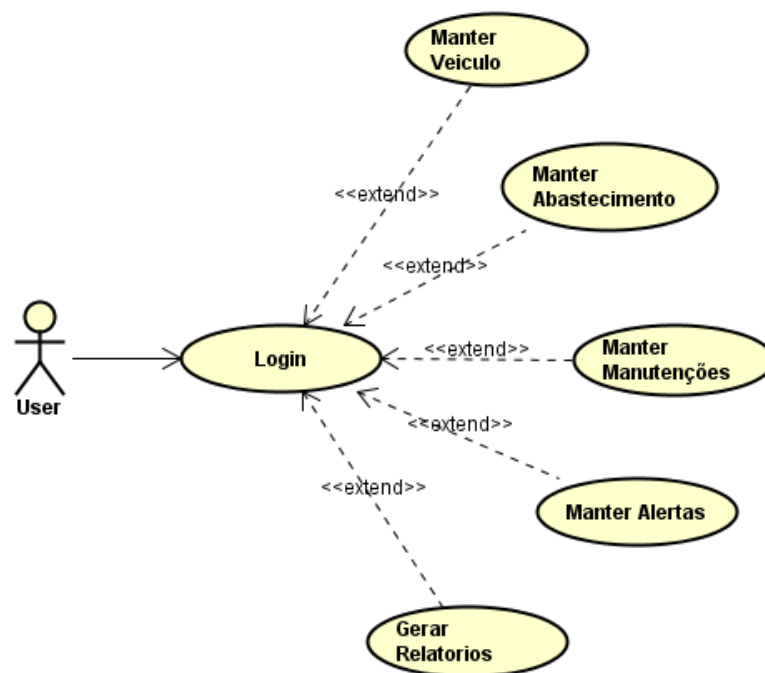
sugerir a manutenção a partir de uma determinada quilometragem realizada ou dentro de um determinado período de tempo, por exemplo a cada 12 meses. Para auxiliar o usuário nesse controle, o aplicativo prevê a configuração de alertas, com dados da manutenção a ser realizada. Dessa forma, a tela inicial do aplicativo vai mostrar as manutenções previstas a curto prazo. Também apresentará os valores totais dos gastos do usuário para o mês corrente.

### 3.1.1 Diagrama de caso de uso

O diagrama de casos de uso tem o objetivo de mostrar a interação do sistema com os seus atores, assim, a Figura 4 representa os casos de uso da aplicação proposta.

Para acessar a aplicação o usuário necessita logar-se no sistema, utilizando seu e-mail e senha para autenticação. Após autenticação, o usuário tem acesso as seguintes funcionalidades: manter veículos, manter abastecimentos, manter manutenções, manter alertas e gerar relatórios.

Figura 4 – Diagrama de casos de uso do aplicativo proposto

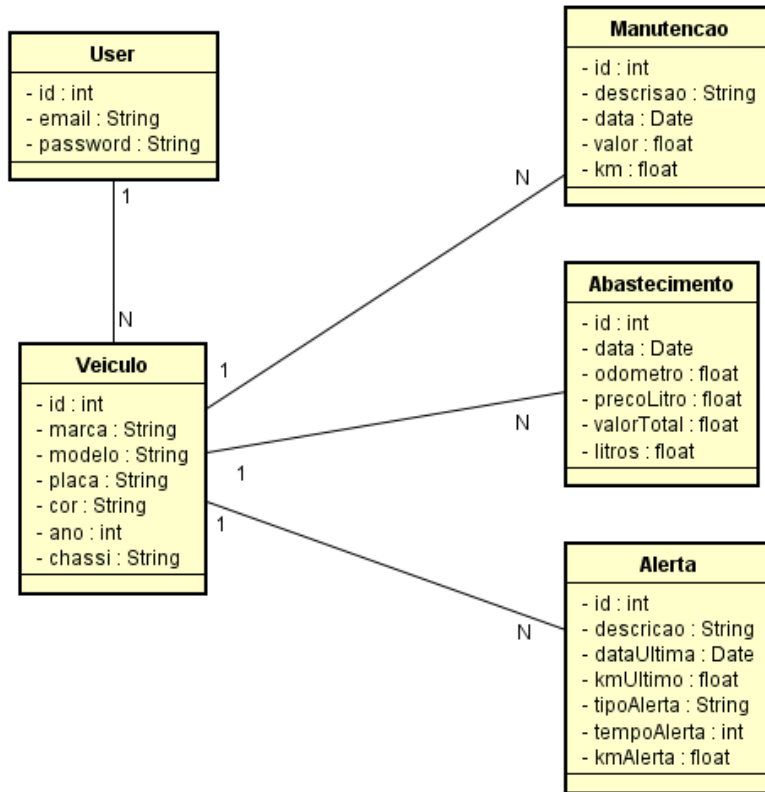


Fonte: Autor, 2021

### 3.1.2 Diagrama de classes

O diagrama de classe na Figura 5 é a representação da persistência dos dados gerenciados pela aplicação, contendo a relação entre as classes.

Figura 5 – Diagrama de classes do aplicativo proposto

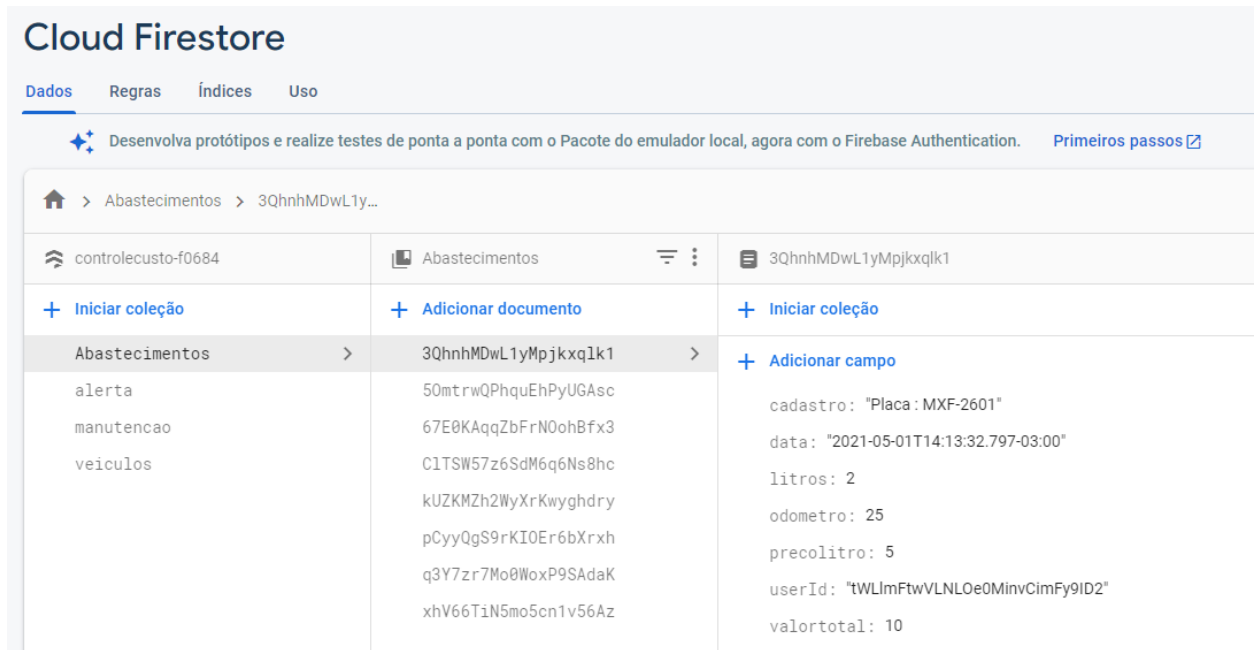


Fonte: Autor, 2021

### 3.2 Persistência de dados

A estrutura da aplicação no banco de dados Cloud Firestore é composta por quatro coleções: abastecimentos, alerta, manutencao e veiculos), conforme Figura 6.

Figura 6 – Estrutura do Cloud Firestore



Fonte: Do autor, 2021

### 3.3 Desenvolvimento do App

Para o desenvolvimento do PWA foi utilizado o *framework* Ionic. Cabe destacar que o Ionic, a partir do desenvolvimento utilizando a sua estrutura, permite a geração da aplicação para a abordagem PWA.

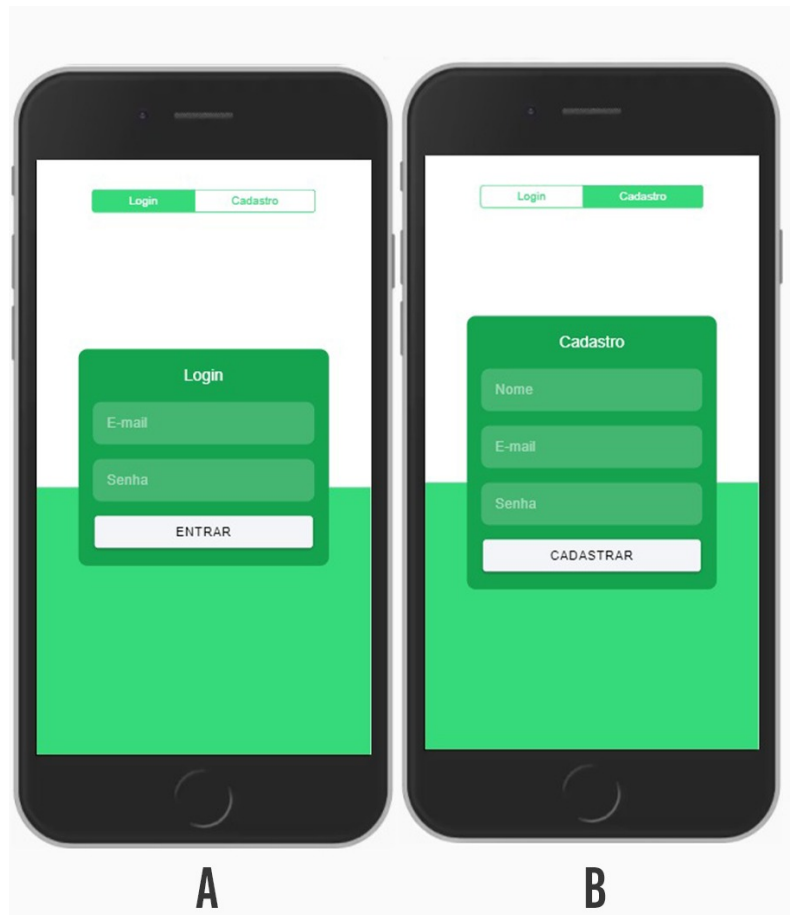
Para persistência dos dados foi utilizada a plataforma Firebase com o Cloud Firestore, ou seja, uma solução de armazenamento remoto. Essa solução foi definida para garantir que os dados sejam preservados, algo que não seria possível se o armazenamento fosse local e ocorresse algum problema com o dispositivo do usuário.

O aplicativo pode ser utilizado por vários usuários e cada usuário podendo assim acessar exclusivamente as suas informações mediante autenticação.

#### 3.3.1 Login e Cadastro

Para poder acessar o aplicativo, foi utilizado o recurso do Firebase Authentication, que gera um identificador de cada usuário registrado, sendo assim o usuário necessita realizar um cadastro no aplicativo para poder então fazer o login. A Figura 7A apresenta a tela de login e a Figura 7B a de cadastro.

Figura 7 – Tela de login e cadastro



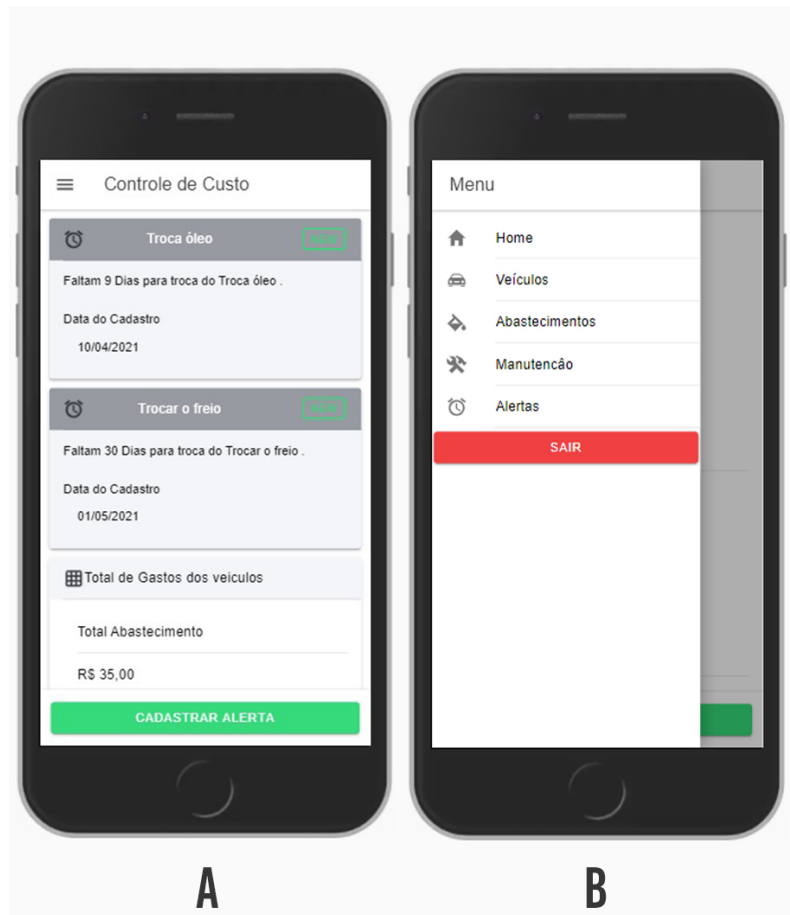
Fonte: Do autor, 2021

### 3.3.2 Tela Inicial e Menu

A página inicial da aplicação, mostrada após a autenticação do usuário, apresenta importantes informações. Como pode ser visto na Figura 8A, na parte superior da página são listadas os alertas, caso esteja próximo ao período configurado para realização da manutenção. Por exemplo, caso exista uma alerta configurada por tempo, é calculado o número de dias que faltam para sua realização com base na data da última manutenção + número de meses \* 30. Assim, se em relação a data atual, faltarem menos de 30 dias, a alerta é mostrada para o usuário, possibilitando assim que ele se programe para a sua realização. Abaixo são listados os totais para abastecimentos e manutenções realizados dentro do mês corrente.

O menu da aplicação (Figura 8B) é acessível por meio do botão sanduíche do lado esquerdo. Quando pressionado apresenta as opções: Home que apresenta a tela inicial; Veículo que remete para listagem de veículos do usuário; Abastecimentos com lista os abastecimentos cadastrados; Manutenções onde estão relacionados todas as manutenções registradas; Alertas que possui a listagem das configurações de alertas; Sair que faz o logout da aplicação.

Figura 8 – Tela inicial e menu



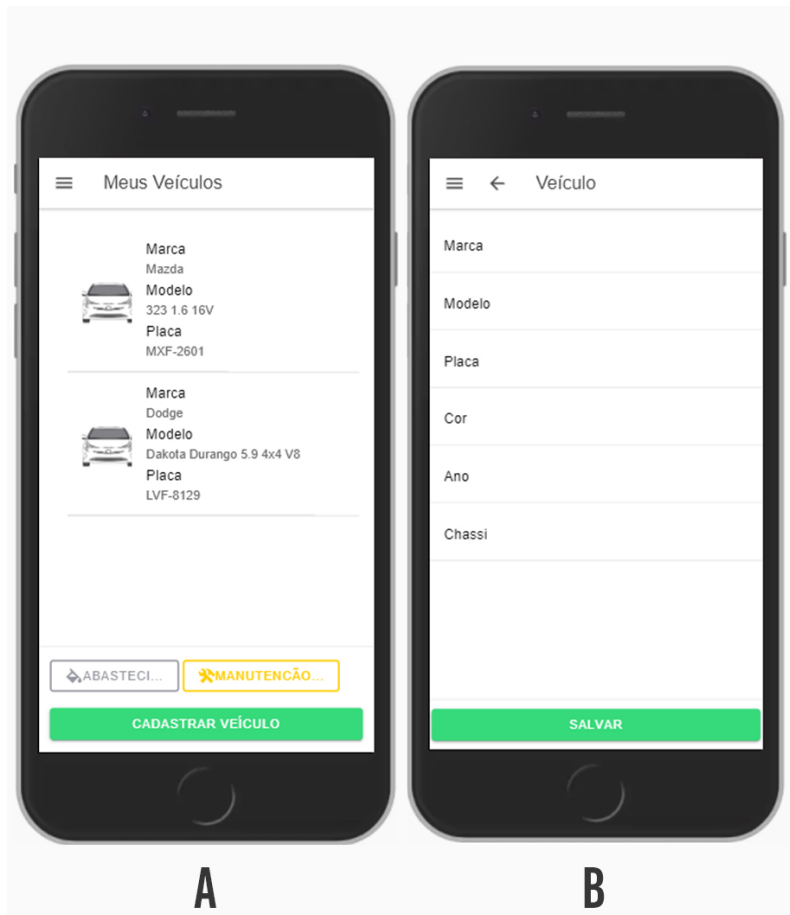
Fonte: Do autor, 2021

### 3.3.3 Telas dos Veículos

Quando o usuário seleciona a opção “Veículos” do menu, é apresentada tela da Figura 9A com a lista de todos os veículos do usuário. Nessa tela estão disponíveis três botões. O botão “Cadastrar veículo” direciona para a tela de cadastro, além dos botões para direcionar o usuário para abastecimentos e manutenções. Para fazer uma alteração de um veículo o usuário deve clicar nos dados do veículo na listagem. Já para fazer a exclusão o usuário deve deslizar os dados do veículo para esquerda, assim será apresentado um botão para realizar a operação de exclusão do veículo.

Na página para realizar o cadastro do veículo, conforme Figura 9B, o usuário deve informar os dados e clicar no botão “Salvar” para efetivar a gravação. Todos os campos do formulário deveram ser preenchidos, caso haja tentativa de salvar sem ter todos os campos preenchidos, aparecerá uma mensagem que todos os campos são obrigatórios.

Figura 9 – Veículos



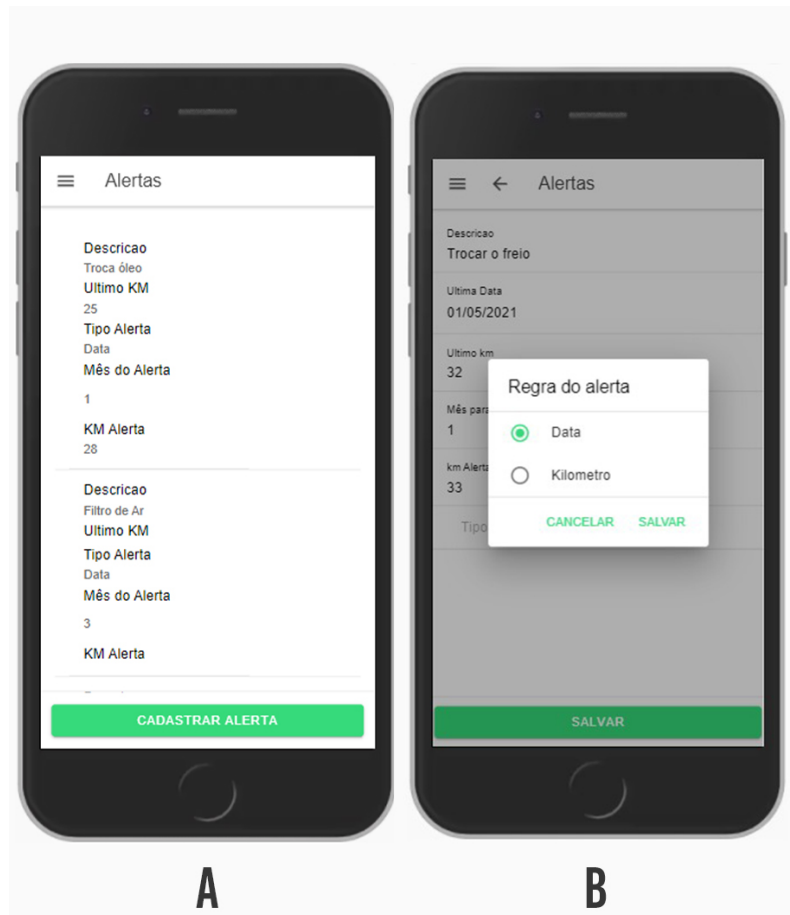
Fonte: Do autor, 2021

As demais manutenções da aplicação (abastecimentos, manutenções e alertas) seguem o mesmo padrão da manutenção de veículos.

### 3.3.4 Alertas

Na tela de Alertas, mostrada na Figura 10A, estão listadas todos os registros de alertas configurados pelo usuário. Ao configurar um alerta o usuário informa: o veículo, descrição do alerta, data da última realização, quilometragem da última realização, regra do alerta (tempo ou km), informar número de meses para tipo tempo e quilometragem para alerta no caso de tipo de alerta km. A Figura 10B apresenta a tela de cadastro na seleção do tipo de regra do alerta.

Figura 10 – Alertas



Fonte: Do autor, 2021

## Considerações finais

Como citado anteriormente, manter um veículo envolve várias manutenções e custos, mas todos necessários para garantir que estejam em perfeito estado. Além do custo habitual com combustível, o carro possui um desgaste natural das suas peças, o que leva a manutenção corretiva, onde é feita a troca de peças e/ou reparos. De outro lado, a manutenção preventiva deve seguir o calendário determinado no manual do veículo. Nesse processo, pode ser necessário a troca de componentes ou fazer ajustes, inspeções, entre outras verificações. Esse tipo de manutenção contribui para a duração dos componentes, e pode reduzir custos no futuro.

Este trabalho apresentou a proposta de um PWA que possibilita ao usuário controlar os custos dos seus veículos e gerenciar futuras manutenções. A opção de desenvolver utilizando a metodologia PWA permite que a aplicação seja executada pelo navegador, dando ao usuário de dispositivo móvel a experiência próxima de operar um aplicativo instalável. O Framework Ionic foi usado para desenvolvimento e para a persistência dos dados foi adotado o serviço Cloud Firestore da plataforma Google Firebase. Além disso, também foram utilizados os serviços de autenticação e

hospedagem do Firebase. Com a abordagem PWA associada ao Firestore, a solução também tem potencial para operar de forma off line.

A solução permite ao usuário cadastrar vários veículos, lançar os abastecimentos e manutenções, além de configurar alertas para futuras manutenções, que são apresentadas na tela inicial do aplicativo juntamente com o total de abastecimentos e manutenções do mês.

Tem-se como proposta de melhorias futuras a possibilidade de adicionar e implementar os alertas por quilometragem, os relatórios e gerenciar a versão do aplicativo para informar ao usuário.



# A PWA TO MANAGE VEHICLE COSTS USING IONIC

Daniel Granville dos Santos\*  
Anubis Graciela de Moraes Rossetto †

2021

## Abstract

This work presents a progressive web application (PWA) to control user's vehicle costs. The idea is to provide the user with an agile way of recording through their mobile device all the costs that are generated by their vehicle. In addition, with some information provided by the user, the app can alert to the need for some type of maintenance, for example, oil change, filter change. For the development, we opted to use the Ionic framework that allows the creation of hybrid multiplatform applications, being adopted for this project the generation of a PWA that has as a characteristic to look like a native application, although it is connected in the browser. For data persistence, a Cloud Firestore platform, a remote storage solution, was used.

**Keywords:** vehicle cost. PWA. Ionic.

## Referências

BISPO, D. M. et al. Desenvolvimento de jogo educacional sobre ecotoxicologia utilizando html5. *Revista Brasileira de Informática na Educação*, v. 20, n. 1, p. 121, 2012. Citado na página 4.

---

\*<dani17pf@gmail.com>

†<anubisrossetto@ifsul.edu.br>

DRIVVO. *Drivvo*. 2021. Disponível em: <<https://www.drivvo.com/pt/>>. Acesso em: 26 abr 2021. Citado na página 6.

FIREBASE. *Cloud Firestore*. 2021. Disponível em: <<https://firebase.google.com/docs/firestore?hl=pt-br>>. Acesso em: 25 abr 2021. Citado na página 5.

FIREBASE. *Firebase Authentication*. 2021. Disponível em: <<https://firebase.google.com/docs/auth?hl=pt-br>>. Acesso em: 25 abr 2021. Citado na página 6.

FIREBASE. *Firebase Hosting*. 2021. Disponível em: <<https://firebase.google.com/docs/hosting?hl=pt-br>>. Acesso em: 25 abr 2021. Citado na página 6.

FIREBASE. *Firebase por plataforma*. 2021. Disponível em: <<https://firebase.google.com/docs?hl=pt-br>>. Acesso em: 25 abr 2021. Citado na página 5.

FIREBASE. *Uso e limites*. 2021. Disponível em: <<https://firebase.google.com/docs/firestore/quotas?hl=pt-br>>. Acesso em: 25 abr 2021. Citado na página 5.

FLANAGAN, D. *JavaScript: O guia definitivo*. [S.l.]: Bookman Editora, 2004. Citado na página 5.

FUELIO. *Aplicativo Fuelio*. 2021. Disponível em: <<https://www.fuel.io/>>. Acesso em: 26 abr 2021. Citado 2 vezes nas páginas 7 e 8.

GRILLO, R. *Introdução ao Ionic Framework*. 2017. Citado na página 4.

IONICFRAMEWORK. *Componentes de interface do usuário*. 2019. Disponível em: <<https://ionicframework.com/docs/components>>. Acesso em: 22 abr 2021. Citado na página 5.

LUIZ, W. *O que são Progressive Web Apps?* 2021. Disponível em: <<https://web.dev/what-are-pwas/>>. Acesso em: 20 abr 2021. Citado na página 2.

MICROSOFT artigo da. *Welcoming Progressive Web Apps to Microsoft Edge and Windows 10*. 2021. Disponível em: <<https://blogs.windows.com/msedgedev/2018/02/06/welcoming-progressive-web-apps-edge-windows-10/>>. Acesso em: 25 abr 2021. Citado na página 3.

RUSSELL, A. *Progressive Web Apps Escaping Tabs Without Losing Our Soul*. 2015. Disponível em: <<https://infrequently.org/2015/06/progressive-apps-escaping-tabs-without-losing-our-soul/>>. Acesso em: 25 abr 2021. Citado na página 3.

SANTOS. *Calcular Combustível*. 2021. Disponível em: <<https://play.google.com/store/apps/details?id=br.com.jurandysantos.combustivel>>. Acesso em: 26 abr 2021. Citado na página 7.

SILVA, M. *Quais são os principais frameworks para desenvolver aplicações móveis híbridas?* 2018. Disponível em: <<https://medium.com/@mattlack/quais-s~ao-os-principais-frameworks-para-desenvolver-aplicac~oes-moveis-hibridas-2093b64d679e>>. Acesso em: 22 abr 2021. Citado na página 5.

VIEBRANTZ, A. F. P. M.; CAMPOS, G. F. da S. Construindo aplicativos híbridos com ionic framework. *Tendências e Técnicas em Sistemas Computacionais*, p. 60, 2015. Citado na página 4.