



UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

ESCOLA DE INFORMÁTICA APLICADA

Proteste Postos:

um aplicativo móvel utilizando o cross-framework Phonegap

Bernardo Salgueiro

Orientador

Mariano Pimentel

RIO DE JANEIRO, RJ – BRASIL

DEZEMBRO DE 2014

Proteste Postos:
um aplicativo móvel utilizando o cross-framework Phonegap

Bernardo Salgueiro

Projeto de Graduação apresentado à Escola de
Informática Aplicada da Universidade Federal do
Estado do Rio de Janeiro (UNIRIO) para obtenção do
título de Bacharel em Sistemas de Informação.

Aprovada por:

Mariano Pimentel (UNIRIO)

Morganna Diniz (UNIRIO)

José Ricardo Cereja (UNIRIO)

RIO DE JANEIRO, RJ – BRASIL.

DEZEMBRO DE 2014

Agradecimentos

Dedico este trabalho a todos os que fizeram parte da minha vida acadêmica. Todos eles contribuíram de alguma forma para a realização deste trabalho. Em especial, Ricardo dos Santos e Thaíssa Diirr pela companhia nesta jornada e o professor Mariano Pimentel, por não ter desistido de mim.

RESUMO

Este trabalho apresenta e descreve o aplicativo para smartphones Proteste Postos, desenvolvido pela Proteste: associação de defesa do consumidor, que auxilia o usuário a achar postos de gasolina e comparar os seus preços. O aplicativo foi criado utilizando o framework híbrido Phonegap e este trabalho descreve suas características, vantagens e desvantagens em relação ao desenvolvimento nativo de aplicações mobile.

Palavras-chave: desenvolvimento, smartphone, framework híbrido, Phonegap.

ABSTRACT

This paper presents and describes the application Proteste Postos for smartphones, developed by Proteste: consumer protection association, which helps the user to find gas stations and compare their prices. The application was created using the cross-framework Phonegap and this paper describes its characteristics, advantages and disadvantages compared to the development of native mobile applications.

Keywords: development, smartphone, cross-framework, Phonegap.

Índice

1 Introdução	7
1.1 Objetivo.....	7
1.2 Justificativa do projeto	7
1.3 Organização do texto.....	9
2 Desenvolvendo para smartphones.....	11
2.1 O mercado de smartphones	11
2.2 Desenvolvendo aplicações para smartphones	12
2.3 Explicando os Frameworks Híbridos	13
2.4 Frameworks Híbridos: Vantagens e desvantagens.....	14
3 Explicando o aplicativo.....	16
3.1 Sobre o aplicativo.....	16
3.2 Explicando o Phonegap.....	16
3.3 Arquitetura do Software	18
3.4 Casos de uso.....	20
3.5 Diagrama de Classes	23
3.6 Diagrama de Banco de Dados	23
3.7 Back Office	24
3.8 Distribuição e histórico de downloads	24
4 Descrição das telas e funcionalidades.....	26
5 Conclusão.....	37

Índice de Tabelas

Tabela 1 - Comparação entre linguagens de desenvolvimento para cada sistema operacional.....	13
Tabela 2 - Comparativo entre desenvolvimento nativo e desenvolvimento híbrido	15

Índice de Figuras

Figura 1 - Venda de smartphones na América Latina	11
Figura 2 - Comparativo entre Desenvolvimento Nativo e Cross-Platform	14
Figura 3 - Explicando o processo de desenvolvimento utilizado o Phonegap	17
Figura 4 - Diagrama de componentes da aplicação Proteste Postos.....	19
Figura 5 - Principais casos de uso da aplicação Proteste Postos	20
Figura 6 - Diagrama de classes da aplicação Proteste Postos.....	23
Figura 7 - Diagrama de banco de dados da aplicação Proteste Postos	24
Figura 8 - Splash screen	26
Figura 9 - Tela de listagem.....	27
Figura 10 - Tela de mapa.....	28
Figura 11 - Tela de detalhes do posto.....	29
Figura 12 - Tela de simulação de abastecimento.....	30
Figura 13 - Tela de rota até o posto com mapa	31
Figura 14 - Tela de rota com descrição textual até o posto	32
Figura 15 - Tela de busca	33
Figura 16 - Tela de perfil do usuário (1/2)	34
Figura 17 - Tela de perfil do usuário (2/2)	35
Figura 18 - Tela de informações institucionais	36

1 Introdução

1.1 Objetivo

Este trabalho tem como objetivo descrever a aplicação Proteste Postos para sistemas operacionais móveis iOS e Android, cuja finalidade é detectar, através de geoposicionamento, os postos de gasolina mais próximos do usuário e listar seus preços e tipos de combustíveis disponíveis. Dessa forma, o usuário pode decidir a melhor opção na hora de abastecer o seu veículo, seja onde for.

Para este software, foram aplicados os conceitos de desenvolvimento de sistemas para aparelhos móveis e também de sistemas para a web, dada a utilização de frameworks multi-plataforma baseado nos padrões da web como HTML, CSS e javascript. Para o back-end da aplicação, foram aplicados os conceitos de desenvolvimento de aplicações web e serviços web (também conhecidos como web services) que são responsáveis por prover os dados para o aplicativo de fato. Além disso, o back-end também é responsável por coletar e processar toda a informação disponível no aplicativo.

1.2 Justificativa do projeto

Nos últimos anos, houve um grande crescimento no número de pessoas que adotaram plataformas móveis para o uso cotidiano. Smartphones e tablets se popularizaram mundialmente e viraram a grande tendência tecnológica dos últimos anos. Estes aparelhos não só ganharam espaço entre os computadores pessoais como também vieram, muitas vezes, a substituir o desktop. Muitas tarefas cotidianas deixaram de ser realizadas apenas nos computadores tradicionais e passaram a ser realizadas nos dispositivos móveis. Checar e-mails, redes sociais, notícias, entre muitas outras tarefas, passaram a ser possíveis de serem realizadas em qualquer lugar, a qualquer hora.

Segundo dados estatísticos do Google, já existem mais smartphones conectados à internet do que computadores pessoais (Moreira, E, 2012). Tais dados mostram que a computação móvel realmente veio para ficar e representa uma nova era na computação: smartphone é o novo computador pessoal.

Mas os dispositivos móveis, não são tão novos em termos de tecnologia. Já vimos, na década de 90, algumas empresas apostarem em aparelhos portáteis como uma nova solução de tecnologia para sua época. Alguns fatores culminaram na não popularização de tais plataformas no passado, como a falta de uma infraestrutura para dar suporte à comunicação entre estes dispositivos e a internet, e de meios de distribuição que incentivassem o desenvolvimento de aplicações. Tais fatores já foram superados e outro fator importante para o atual sucesso foi a mudança do público alvo dos dispositivos móveis, antes restrito, se tornou um fenômeno social.

A popularização dos novos dispositivos fez a tecnologia avançar rapidamente. Com a mobilidade e a portabilidade, vieram também os sensores integrados, como GPS, acelerômetro, câmera, entre outros, que possibilitaram aos desenvolvedores construir aplicações que aproveitassem esses dados para proporcionar uma nova forma de interação com o usuário. Surgiram aplicações baseadas no contexto do usuário: dados como localização, movimentação, e imagens são o diferencial ao comparar com aplicações desktop. Essas novas aplicações eram impensáveis no computador pessoal por não ter os sensores presentes nos dispositivos móveis.

A popularização dos aparelhos portáteis fez surgir cada vez mais plataformas no mercado, trazendo consigo novos sistemas operacionais. Além disso, trouxe também novos paradigmas e desafios a serem encarados pelos desenvolvedores, que se viram na necessidade de adquirir conhecimento necessário para suprir rapidamente a demanda por aplicativos.

Cada sistema operacional possui frameworks, linguagens de programação, padrões de desenvolvimento próprios e aprender a desenvolver para uma nova plataforma e suas peculiaridades geralmente é um processo que consome tempo. No cenário atual, onde existem vários sistemas operacionais móveis consolidados, é ainda mais demorado criar e manter aplicações para todos eles. No caso de uma empresa, é caro manter profissionais para lidar com cada um dos sistemas operacionais alvos de suas aplicações.

Para resolver este problema, surgiu o que hoje é conhecido como frameworks híbridos, ou do inglês cross-frameworks, que são frameworks que reduzem o trabalho

dos desenvolvedores e reaproveitam o mesmo código, ou pelo menos a maior parte dele, usando-o nas diferentes plataformas já estabelecidas no mercado. Dessa forma os desenvolvedores não precisam conhecer as linguagens, ferramentas e metodologias específicas para desenvolver em cada plataforma. O código é reaproveitado e traduzido pelo framework para as plataformas nativas, gerando assim, a mesma experiência para os usuários das diferentes plataformas.

Nos frameworks híbridos são utilizados os mesmos padrões de desenvolvimento para a web. Linguagens como HTML, CSS e Javascript são amplamente utilizadas na construção de aplicações multi-plataforma. Isso acaba sendo outra grande vantagem para os desenvolvedores, pois ao invés de terem que aprender uma nova linguagem de programação, baseiam-se em padrões já conhecidos e previamente utilizados.

Além das vantagens, existem desvantagens em comparação com o desenvolvimento de aplicações nativas em seus sistemas operacionais. Performance prejudicada e limitação no acesso dos recursos dos aparelhos podem ser consideradas exemplos de desvantagens na utilização dos cross frameworks. Cabe aos desenvolvedores analisarem as vantagens e desvantagens e decidirem se vale a pena adotar esta solução em seus projetos.

Neste trabalho foi utilizado um framework híbrido no desenvolvimento do aplicativo apresentado na próxima seção. Neste projeto, o cross framework mostrou-se útil por reduzir o conhecimento necessário para construir a aplicação para mais de um sistema operacional, e por possibilitar o desenvolvimento por uma equipe reduzida sem profissionais especializados em cada sistema operacional.

1.3 Organização do texto

O presente trabalho está estruturado em capítulos e, além desta introdução, será desenvolvido da seguinte forma:

- Capítulo II: neste capítulo descrevemos o mercado de smartphones, o desenvolvimento para os sistemas operacionais mobile e seus desafios.
- Capítulo III: este capítulo descreve a parte técnica da aplicação.
- Capítulo IV: este capítulo apresenta as funcionalidades e telas da aplicação.

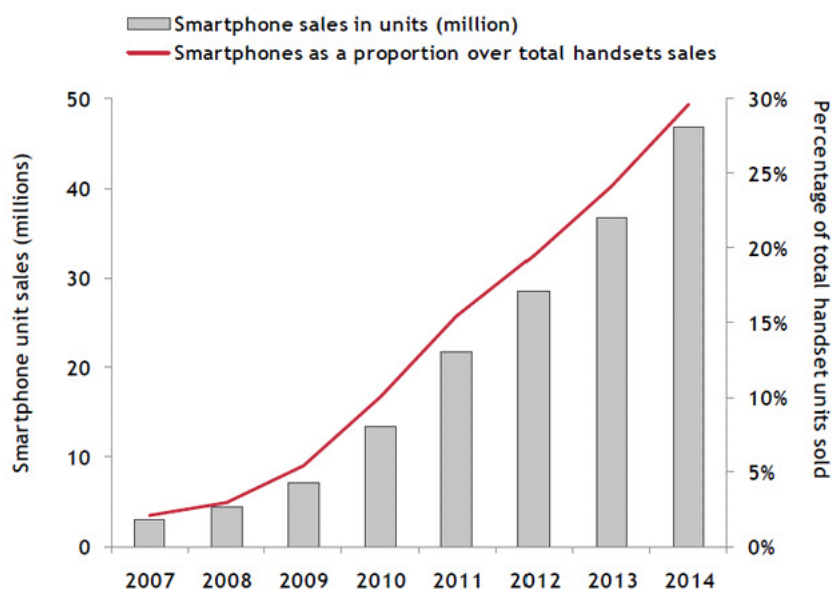
- Capítulo V: Conclusões – Reúne as considerações finais, assinala as contribuições da pesquisa e sugere possibilidades de aprofundamento posterior.

2 Desenvolvendo para smartphones

2.1 O mercado de smartphones

O celular é o novo computador pessoal. O mercado de smartphones cresceu muito nos últimos anos e continua a crescer cada vez mais rapidamente, como é possível ver na figura 1. Os telefones estão sendo usados para atividades cotidianas, assim como são utilizados os computadores. Smartphones são geralmente mais baratos do que computadores, mais convenientes devido à sua portabilidade, e muitas vezes mais versáteis do que um laptop, devido à capacidade de seus sensores, câmera integrada, conexão, geolocalização e portabilidade.

Exhibit: Smartphones as a proportion of total handset unit sales in Latin America and globally, 2008-2014



Source: Pyramid Research Handsets Forecast, Q2 2009

Figura 1 - Venda de smartphones na América Latina
(<http://www.pyramidresearch.com/smartphone-forecast.htm>)

Atualmente, o mercado de Smartphones se divide em dois principais grandes Sistemas Operacionais: o iOS fabricado pela Apple e pioneiro na criação dos smartphones modernos; e o Android, fabricado pelo Google e adotado por várias empresas em seus dispositivos, responsável por difundir e popularizar os smartphones.

Lançado em 2007 pela Apple, o iPhone conta com mais de 400 milhões de contas de usuários cadastradas, mais de 800 mil aplicativos em sua loja virtual, a App Store, a qual já contabilizou mais de 25 bilhões de downloads desde o seu lançamento (Rasmussen, 2013; Uol, 2012).

O Android, lançado em 2008 é atualmente o sistema operacional móvel mais utilizado do mundo, presente em mais de 2 bilhões de smartphones e tablets que, têm como meio de distribuição de aplicativos a Play Store, contendo mais de 800 mil aplicativos disponível para download. (Iwata, 2013)

Os sistemas operacionais de smartphones inovam-se ativamente para manterem-se atualizados com os avanços de hardware e facilitarem o desenvolvimento com ferramentas melhoradas. Muitas vezes, as inovações mais significativas não são puramente técnicas. Os canais de distribuição de aplicativos, introduzidos com o fenômeno de popularização dos smartphones, reduziram as barreiras para o acesso e o desenvolvimento de aplicações, fornecendo um fácil meio de distribuição.

Ainda que a popularização dos smartphones tenha acontecido a partir de 2007 com o lançamento do iPhone, os dispositivos e aplicações móveis não são novos. No final dos anos 90, o desenvolvimento móvel foi considerado um mercado aquecido. Embora houvesse desenvolvedores de aplicativos independentes e a maioria dos telefones high-end suportassem a instalação de aplicativos, o processo de instalação era difícil e maioria dos usuários finais não adicionavam aplicativos em seus telefones, portanto, não utilizavam outros softwares além dos que já vinham instalados com o aparelho.

2.2 Desenvolvendo aplicações para smartphones

O desenvolvimento de aplicações para sistemas operacionais mobile difere em vários aspectos em relação ao desenvolvimento de aplicações desktop. Além do iOS e Android, existem outros de sistemas operacionais para smartphone no mercado. Além disso, ao contrário dos sistemas operacionais de desktop, o OS em

computação móvel normalmente determina a linguagem de programação que os desenvolvedores devem usar.

Sistema Operacional	Linguagem de programação
Apple iPhone	Objective-C
Google Android	Java
Windows Mobile	C#
RIM BlackBerry	Java
Symbian	C++
Palm webOS	Javascript

Tabela 1 - Comparação entre linguagens de desenvolvimento para cada sistema operacional

Como é possível perceber, para desenvolver para mais de uma plataforma móvel, é necessário conhecimento sobre diversas linguagens, além de seus ambientes de desenvolvimento e frameworks nativos de cada plataforma.

2.3 Explicando os Frameworks Híbridos

Com cada fabricante de dispositivo móvel tendo seu próprio ambiente de desenvolvimento, um crescimento do desenvolvimento de aplicações móveis que são capazes de se comunicar entre si através da internet, e uma grande população de desenvolvedores experientes em HTML, surgiram frameworks de aplicativos baseados na web para ajudar os desenvolvedores a criarem aplicações que podem ser implantados em vários dispositivos, os chamados frameworks híbridos, ou cross-frameworks, ou ainda frameworks multi-plataforma.

Estamos vendo a influência do desenvolvimento web nas emergentes técnicas multi-plataforma para celular. Antes da existência de qualquer framework multi-plataforma, muitos desenvolvedores descobriram que a incorporação de interface de usuário da Web em um aplicativo nativo foi uma forma prática para desenvolver aplicações móveis de forma rápida e fazer aplicações multi-plataforma fácil de manter. A interface de usuário para aplicações móveis tende a ser apresentada como uma série

de telas. A partir de um nível elevado, a interface pode ser pensada como tendo o mesmo fluxo de controle quanto um website tradicional ou aplicação web.

Atualmente existem diversos frameworks cross-platform como Phonegap, Titanium, Rhomobile, Qt Mobility, entre muitos outros. Todos eles têm a mesma premissa de agilizar o desenvolvimento para smartphones utilizando as linguagens da Web como HTML, CSS e Javascript, assim como se beneficiar da reutilização de código entre os sistemas operacionais mobile.

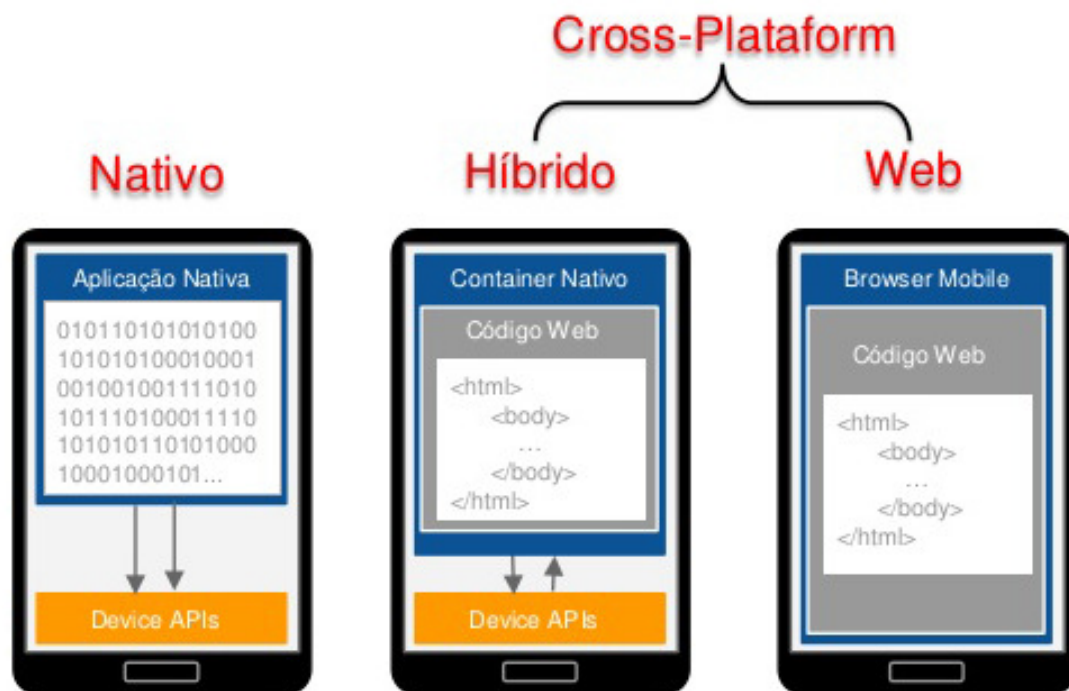


Figura 2 - Comparativo entre Desenvolvimento Nativo e Cross-Platform
(<http://pt.slideshare.net/suelengc/nativo-vs-crossplataform>)

2.4 Frameworks Híbridos: Vantagens e desvantagens

O uso de um framework híbrido, portanto, auxilia os desenvolvedores a portarem suas aplicações para outras plataformas mobile fazendo uso do mesmo código fonte, e por serem baseados em padrões da web já estabelecidos no mercado, eliminam a curva de aprendizado inicial necessária para começar a desenvolver especificamente para cada sistema operacional.

Apesar desses frameworks apresentarem uma alternativa ao desenvolvimento nativo para sistemas operacionais mobile, eles possuem desvantagens em relação ao desenvolvimento nativo. Nem todas as APIs existentes em um ambiente de

desenvolvimento nativo estão disponíveis para os frameworks híbridos, portanto, dessa forma nem sempre é possível ter acesso a todas as funcionalidades que o hardware do dispositivo oferece.

Outra grande desvantagem é em relação à performance: as aplicações híbridas, em relação às aplicações nativas, têm o desempenho prejudicado e podem não apresentar a mesma fluidez vista nas aplicações criadas em seus frameworks nativos. Ainda existe uma questão sobre a identidade visual dos aplicativos, hoje muitos sistemas operacionais móveis possuem guidelines de desenvolvimento que incentivam os desenvolvedores a criarem seus aplicativos seguindo certos padrões visuais, com a justificativa que a experiência para o usuário seja mais familiar e intuitiva. Isso pode ser contornado criando layouts diferentes para cada sistema operacional, porém fere a premissa de reaproveitamento do código e demanda mais tempo dos desenvolvedores.

Funcionalidade	Nativo	Híbrido
Linguagem de desenvolvimento	Específica	Padrão da web
Portabilidade e Reutilização de código	Muito baixa	Alta
Utilização de conhecimento prévio para desenvolvimento	Baixa	Alta
Acesso a recursos de hardware do dispositivo	Alto	Médio
Performance da aplicação	Otimizada	Prejudicada

Tabela 2 - Comparativo entre desenvolvimento nativo e desenvolvimento híbrido

3 Explicando o aplicativo

3.1 Sobre o aplicativo

O aplicativo Proteste Postos é um projeto da organização de defesa do consumidor Proteste, uma ONG voltada para a defesa e orientações de consumo para o consumidor brasileiro. A ideia do aplicativo é indicar os preços dos postos de gasolina mais próximos do consumidor, baseado no geoposicionamento fornecido pelo smartphone do usuário, e listá-los de acordo com a relação custo-benefício, ajudando o consumidor a economizar na hora de abastecer o seu veículo.

O Proteste Postos foi desenvolvido para os sistemas operacionais iOS e Android utilizando do cross framework Phonegap, possibilitando o reaproveitamento de grande parte do código da aplicação na portabilidade entre as plataformas. A escolha deste framework se deve especificamente pelo conhecimento sobre os padrões utilizados, facilidade de uso, e baixo custo de utilização.

3.2 Explicando o Phonegap

O PhoneGap é um framework que permite a criação de aplicações móveis utilizando APIs (Application Programming Interface ou Interface de Programação de Aplicativos) padronizadas da web. Ele foi desenvolvido por uma empresa chamada

Nitobi, que em 2011 foi comprada pela Adobe e utilizando do motor Apache Cordova, criou o Phonegap, que é uma espécie de distribuição comercial deste projeto. O Phonegap é gratuito e é um software livre sob a licença Apache 2.0 (Apache Software Foundation, 2004).

“Para entender melhor, portanto, o PhoneGap é um conjunto de APIs que permite que o desenvolvedor acesse as funções nativas do dispositivos, como câmera, agenda, etc; através de JavaScript, HTML5 e CSS3, em vez de linguagens específicas de dispositivo, como Objective-C. O desenvolvimento é como de qualquer site e por isso oferece uma maior facilidade de aprendizado. Dispensando, por exemplo, longas horas de dedicação a sistemas mais complicados, como Java, além de reduzir custos do projeto.” (Fasciana, 2014)

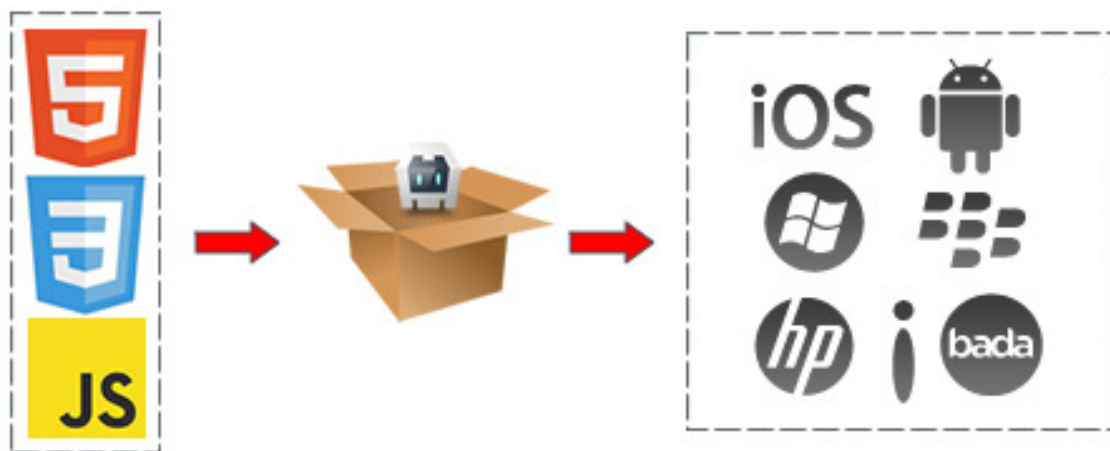


Figura 3 - Explicando o processo de desenvolvimento utilizado o Phonegap
(<http://phonegap.com/2012/05/02/phonegap-explained-visually>)

As aplicações criadas a partir deste framework são híbridas, ou seja, elas não são completamente nativas - porque toda a renderização do layout é feito através de HTML ao invés de utilizar os elementos de interface visual de cada plataforma - nem simplesmente páginas HTML, pois é gerada uma aplicação com acesso às APIs nativas do dispositivo, como citado anteriormente.

3.3 Arquitetura do Software

A arquitetura do Proteste Postos consiste em um aplicativo a ser instalado no smartphone, composto pela parte nativa com as bibliotecas do Phonegap e a parte do front-end que é a parte a ser reaproveitada entre as plataformas, mais os serviços web que constituem o backend. Além disso, há uma outra aplicação responsável por coletar os preços e informações dos postos a partir de um XML, que é gerado a partir de um outro software que faz a coleta dos preços diretamente do site da Agencia Nacional de Petróleo (ANP).

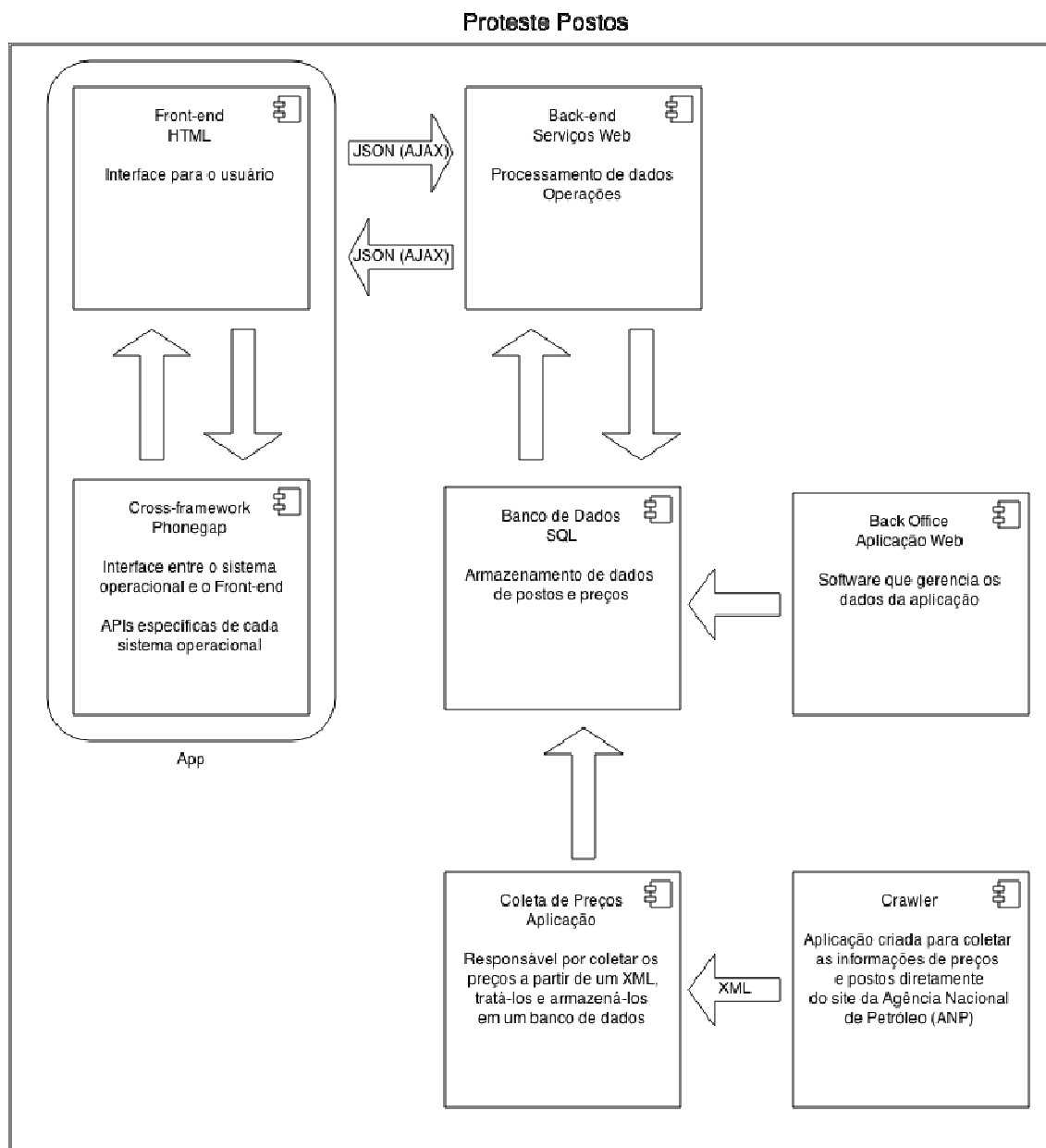


Figura 4 - Diagrama de componentes da aplicação Proteste Postos

Neste diagrama de componentes é apresentado todas as peças que compõe a aplicação Proteste Postos, desde o software responsável pela coleta de preços diretamente da ANP até a apresentação para o usuário através do aplicativo instalado em seu smartphone.

3.4 Casos de uso

Esta seção tem como objetivo descrever em forma de casos de uso as principais funcionalidades da aplicação.

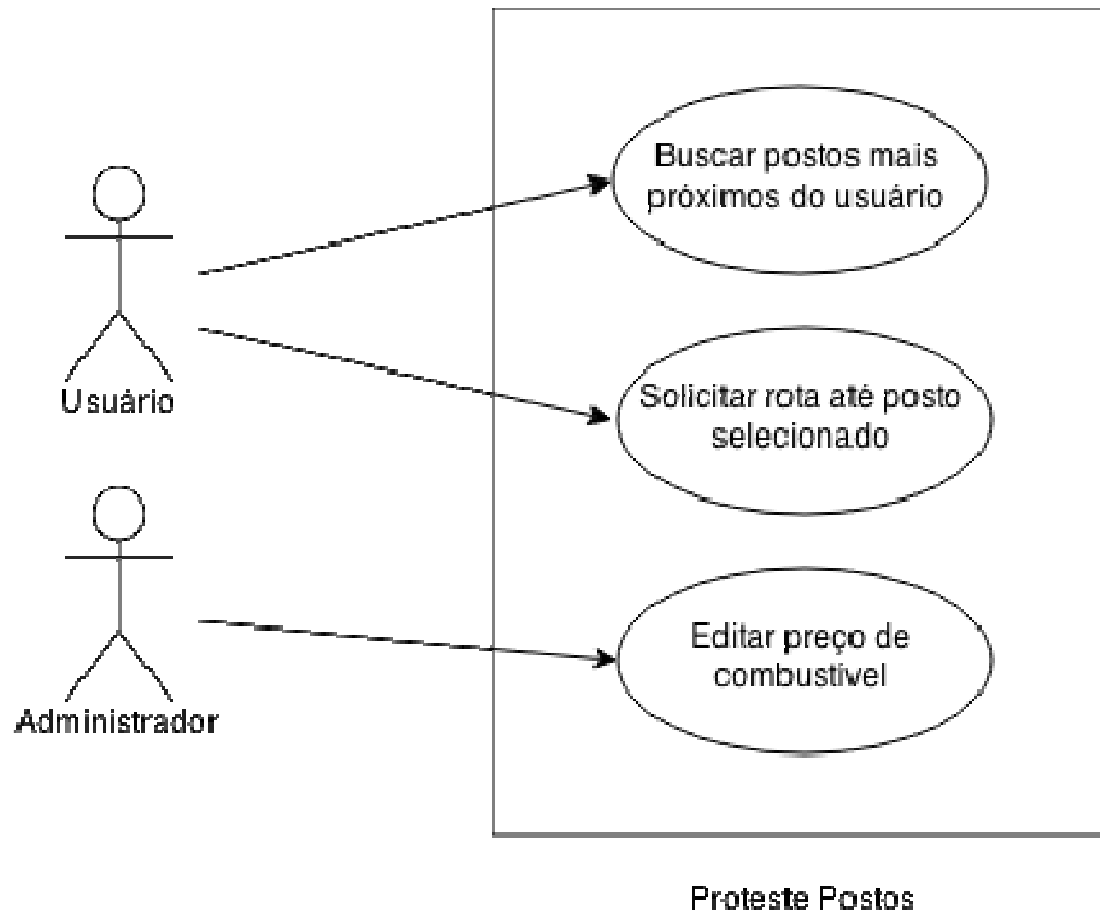


Figura 5 - Principais casos de uso da aplicação Proteste Postos

Caso de Uso 1 – Buscar os postos de gasolina mais próximos do usuário

Descrição: Buscar os postos de gasolina mais próximos do usuário no momento da consulta e ordená-los pela menor distância.

Início: Usuário acessa o sistema

Cenário ideal:

1. O usuário seleciona a opção de busca.

2. Na tela de busca, o usuário seleciona a opção de pesquisar utilizando sua localização atual.
3. O sistema verifica a disponibilidade do sinal de GPS e Internet.
4. O sistema obtém as coordenadas de latitude e longitude do aparelho do usuário.
5. O sistema consulta através dos serviços web os postos mais pertos do usuário baseado nas coordenadas fornecidas pelo aparelho.
6. O sistema retorna para o usuário uma lista com os postos mais próximos com os seguintes atributos: nome, endereço, preços de combustíveis (gasolina, álcool, GNV e diesel), bandeira do posto, distância do usuário.

Caso de Uso 2 – Solicitar rota até posto selecionado

Descrição: Solicitar uma rota a ser exibida em um mapa, fornecendo direções para o usuário chegar até o posto selecionado.

Início: Usuário acessa o sistema

Cenário ideal:

1. O usuário seleciona a opção de lista.
2. O sistema verifica a disponibilidade do sinal de GPS e Internet.
3. O sistema obtém as coordenadas de latitude e longitude do aparelho do usuário.
4. O sistema consulta através dos serviços web os postos mais pertos do usuário baseado nas coordenadas fornecidas pelo aparelho.
5. O sistema retorna para o usuário uma lista com os postos mais próximos com os seguintes atributos: nome, endereço, preços de combustíveis (gasolina, álcool, GNV e diesel), bandeira do posto, distância do usuário.
6. O usuário seleciona um posto exibido na lista.
7. O sistema exibe uma tela com informações do posto selecionado.

8. O usuário seleciona a opção de rota.
9. O sistema exibe uma tela com um mapa e uma rota traçada, descrevendo o caminho do usuário até o posto.

Caso de Uso 3 – Editar preço de combustível

Descrição: O administrador do sistema edita o preço de um posto disponível na aplicação.

Início: Administrador acessa o sistema de back office

Cenário ideal:

1. O administrador seleciona a seção correspondente ao sistema Proteste Postos
2. O sistema apresenta uma tela com a listagem dos postos
3. O administrador seleciona a opção de busca.
4. O administrador entra com o nome do posto cujo preço precisa ser alterado
5. O sistema retorna o resultado de busca com uma lista que apresenta os resultados relevantes às palavras buscadas.
6. O administrador seleciona o posto desejado entre os postos apresentados na lista.
7. O sistema apresenta as informações do posto selecionado, tais como nome, endereço coletado, endereço alternativo, latitude, longitude, e preços dos combustíveis disponíveis.
8. O administrador altera o preço desejado e seleciona a opção “Gravar”.
9. O sistema atualiza a alteração no banco de dados.

3.5 Diagrama de Classes

O diagrama de classe deste projeto representa as entidades que são manipuladas no back-end para gerar a informação a ser enviada para o front-end e, por fim, ser exibida para o usuário.

Foram criadas duas classes: GasStationResult, que representa uma instância de um posto; e GasStationService, que representa a entidade responsável por expor os métodos a serem chamados pela aplicação mobile.

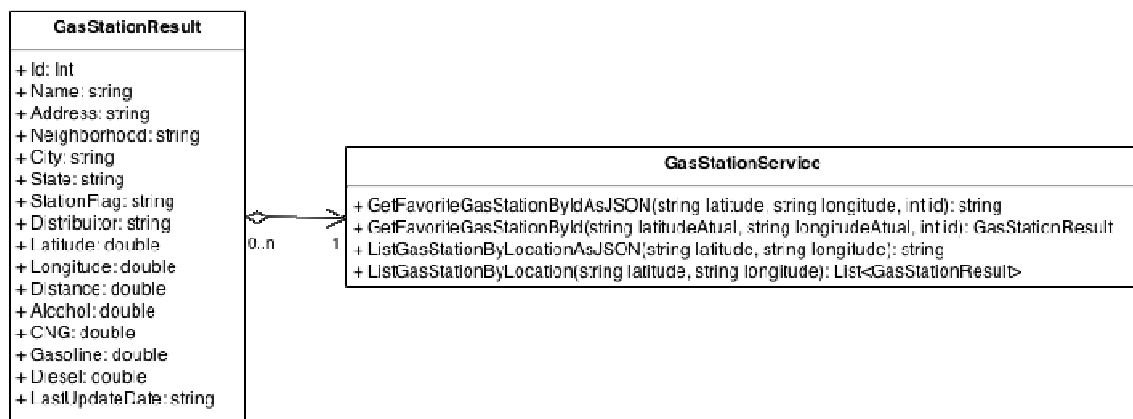


Figura 6 - Diagrama de classes da aplicação Proteste Postos

3.6 Diagrama de Banco de Dados

O banco de dados para esta aplicação consiste em apenas duas tabelas, como é possível ver na figura 7. Uma para representar os postos presentes na aplicação, e outra para representar os seus preços de combustíveis, relacionados através de uma chave estrangeira. É interessante notar que os preços dos postos são instâncias únicas em relação ao tempo, criando assim, um histórico dos preços ao longo do tempo, conforme as informações são atualizadas.

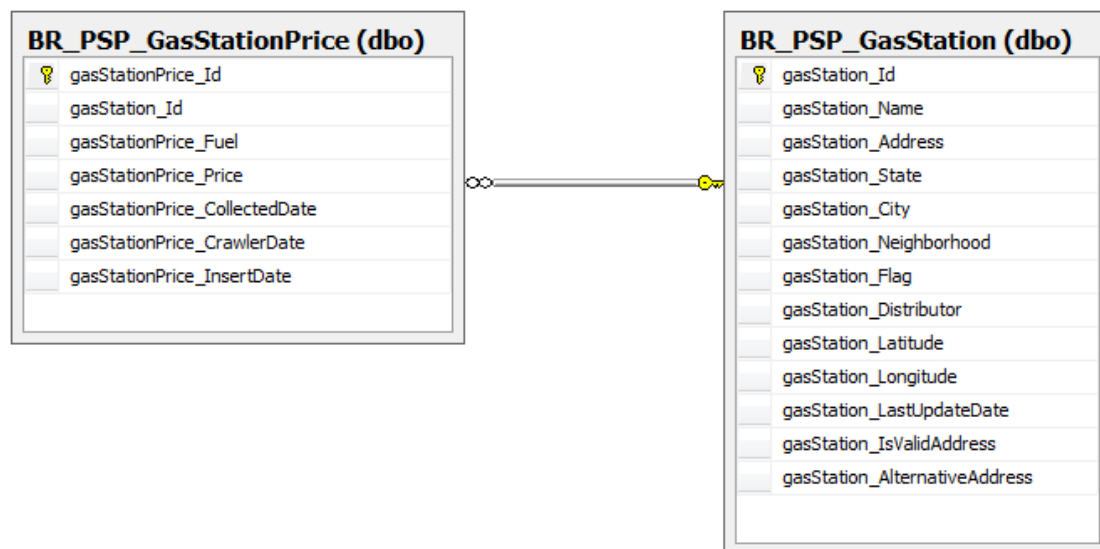


Figura 7 - Diagrama de banco de dados da aplicação Proteste Postos

3.7 Back Office

O Back Office é a aplicação que permite que um administrador do aplicativo altere informações dos postos e seus respectivos preços. É uma aplicação web com acesso ao mesmo banco de dados utilizado pelo back-end.

3.8 Distribuição e histórico de downloads

O aplicativo foi disponibilizado para as plataformas iOS e Android, cada uma com seus canais próprios de distribuição.

No iOS, o app foi disponibilizado pela App Store, loja de aplicativos mantida pela Apple e único canal de distribuição oficial de aplicativos para o sistema operacional. O aplicativo foi lançado no dia 10 de Outubro de 2011 e desde então conta com mais de 20 mil downloads, ficando na posição de aplicativo mais baixado na App

Store em sua semana de lançamento. O aplicativo pode ser encontrado através do link:

<https://itunes.apple.com/br/artist/pro-teste-associacao-brasileira/id421380739>

No Android, o aplicativo foi distribuído através da Google Play Store, principal canal de distribuição de aplicativos para Android, mantido pelo próprio Google.

A versão para Android foi lançada no dia 28 de Fevereiro de 2012 e possui mais de 30 mil downloads. Ela pode ser encontrada através do seguinte link:

<https://play.google.com/store/apps/details?id=com.phonegap.protestepostos>

4 Descrição das telas e funcionalidades

Neste capítulo apresentaremos as principais funcionalidades do aplicativo exibindo suas telas e fazendo uma breve explicação.

1. Splash screen



Figura 8 - Splash screen

Esta é a tela inicial do aplicativo, uma splash-screen que é exibida até o término

do carregamento inicial da aplicação.

2. Tela de Listagem

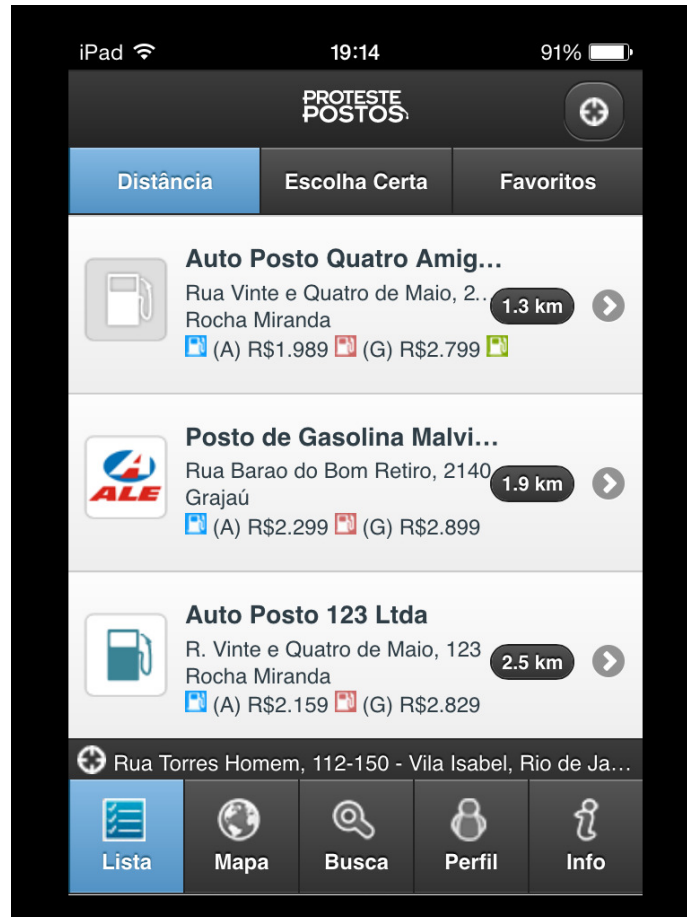


Figura 9 - Tela de listagem

Está é a primeira tela a ser exibida uma vez que o aplicativo termina o seu carregamento. Ela exhibe uma lista com os postos mais próximos do usuário, tendo três opções de ordenação: distância, Escolha-certa (termo próprio da instituição para representar a relação custo-benefício), e mostrar somente os postos marcados como favoritos pelo usuário.

3. Tela de exibição dos postos no mapa

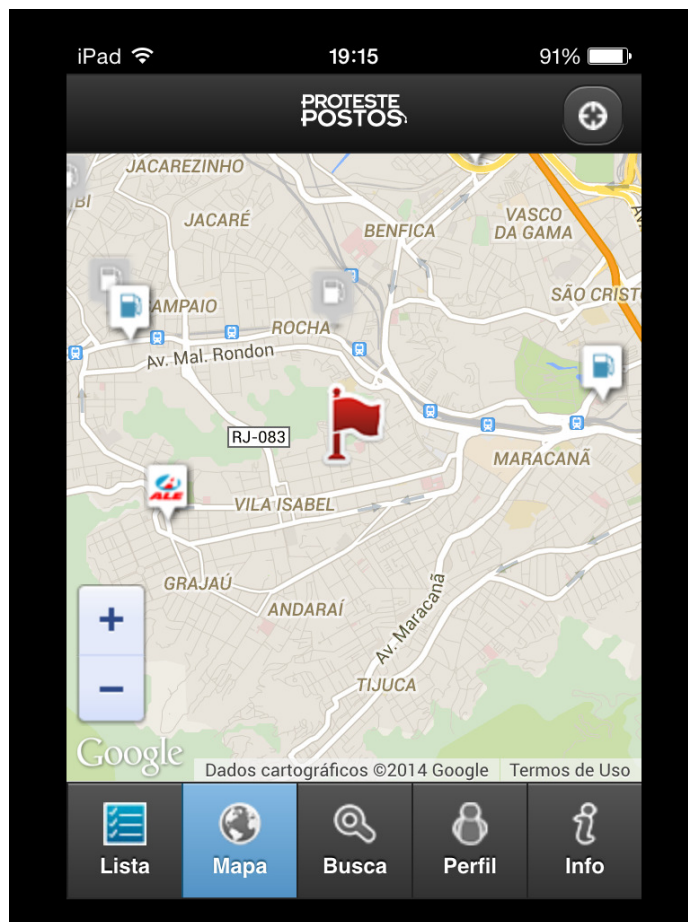


Figura 10 - Tela de mapa

Esta tela exibe a localização do usuário e os postos mais próximos em um mapa.

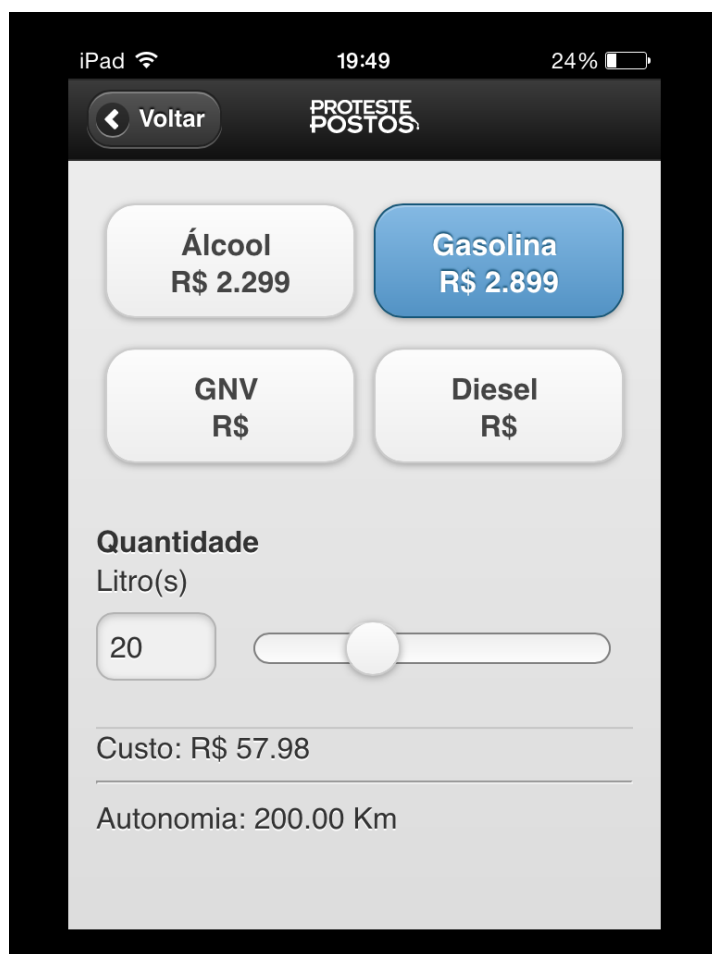
4. Tela de detalhes do posto



Figura 11 - Tela de detalhes do posto

Ao selecionar um posto da lista, é exibida uma tela com os detalhes do posto selecionado, onde é possível realizar quatro ações: Simular um abastecimento com os preços dos combustíveis disponíveis neste posto; Adicionar o posto aos favoritos, Visualizar o posto no mapa e solicitar a rota até este posto.

5. Tela de simulação de abastecimento



Nesta tela, é possível realizar uma simulação de abastecimento, obtendo o custo e autonomia do abastecimento, a partir dos preços fornecidos pela aplicação e os dados do veículo fornecidos pelo usuário.

6. Tela de rota até o posto com mapa

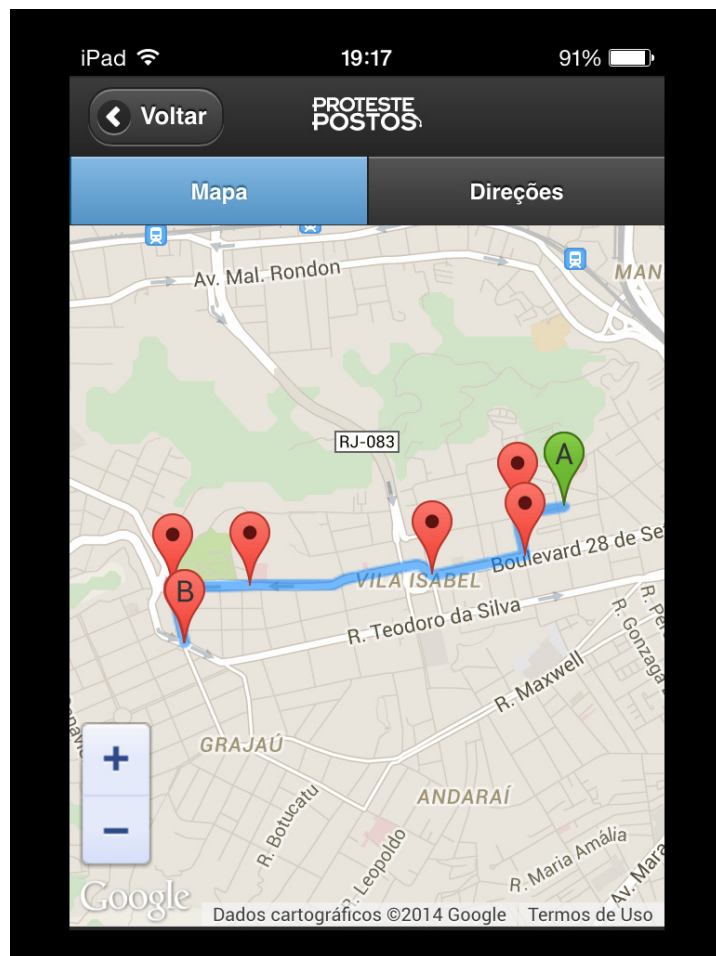


Figura 13 - Tela de rota até o posto com mapa

Nesta tela é exibida uma rota traçada em um mapa com as direções para o usuário chegar da sua localização atual até o posto selecionado.

7. Tela de rota com descrição textual até o posto

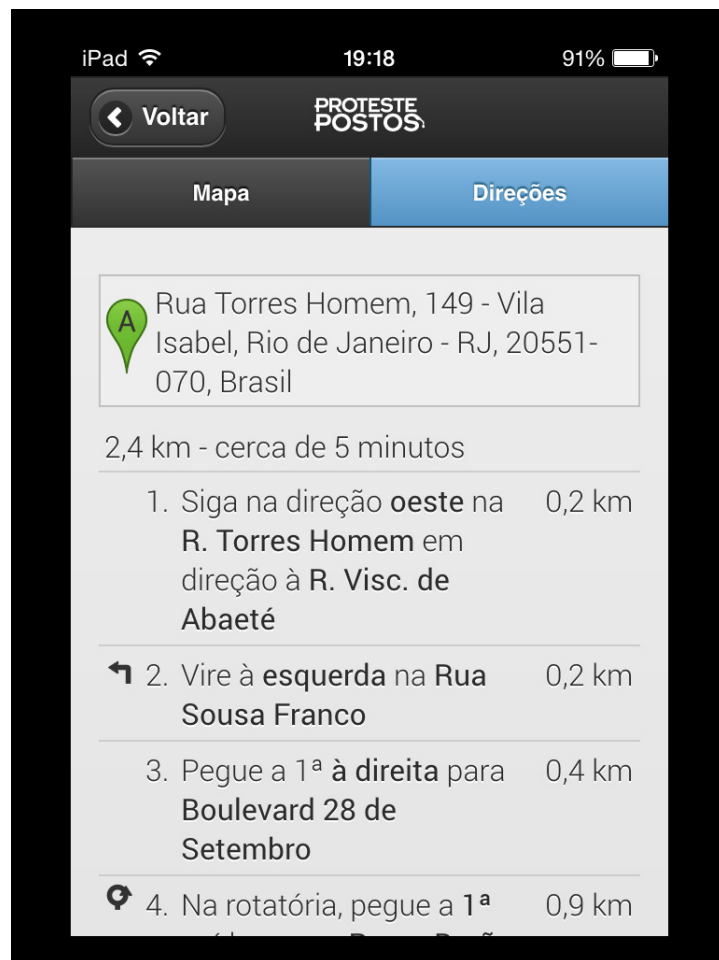


Figura 14 - Tela de rota com descrição textual até o posto

Nesta tela, uma descrição textual da rota é fornecida para o usuário chegar de sua localização atual até o posto selecionado.

8. Tela de busca



Figura 15 - Tela de busca

Nesta tela o usuário pode buscar por postos próximos a um endereço específico fornecido pelo usuário.

9. Tela de perfil do usuário



Figura 16 - Tela de perfil do usuário (1/2)

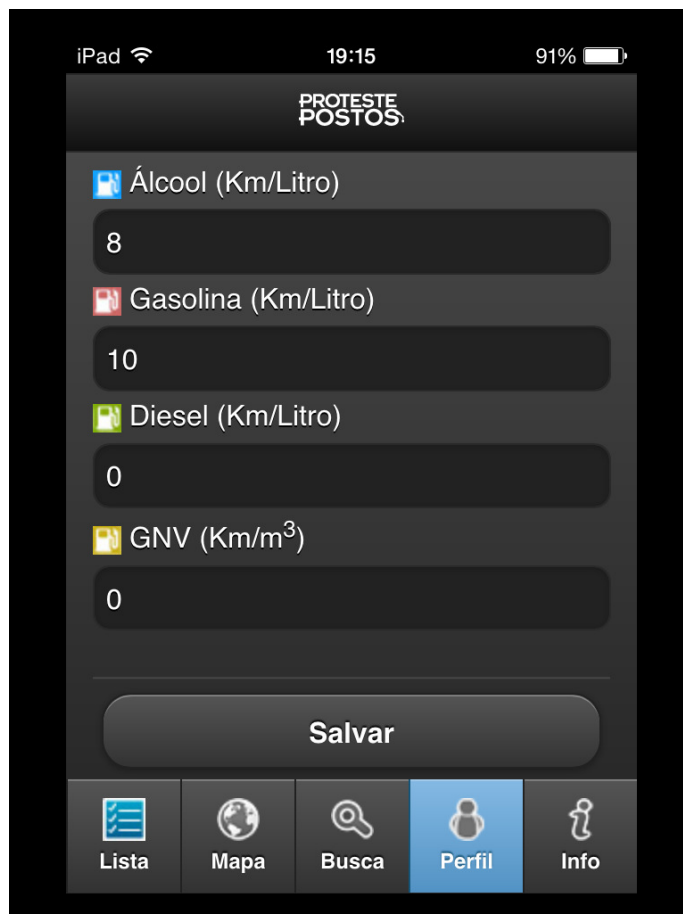


Figura 17 - Tela de perfil do usuário (2/2)

Nesta tela, o usuário pode cadastrar as informações de consumo de seu veículo para serem utilizadas na tela de simulação.

10. Tela de informações institucionais



Figura 18 - Tela de informações institucionais

Nesta tela são fornecidas algumas informações institucionais e de contato.

5 Conclusão

Neste trabalho foi apresentado o aplicativo para smartphones Proteste Postos. Disponibilizado para iOS e Android, ele auxilia o usuário a encontrar os postos de gasolina mais próximos e comparar os preços dos combustíveis disponíveis. Para construir a aplicação, foi utilizado um framework híbrido, que permite a reutilização de código para publicação em diferentes plataformas mobile. Tal framework ainda permite que seja reduzido o tempo de desenvolvimento, uma vez que não é necessário conhecimento técnico específico para desenvolver para cada sistema operacional móvel. Ao invés disso o framework utiliza os padrões de desenvolvimento para a Web, já estabelecidos e conhecidos no mercado.

O trabalho evidencia as vantagens e desvantagens de se utilizar de um framework híbrido para construir uma aplicação mobile. Como principais vantagens, vimos que o uso deste tipo de framework reduz o tempo de desenvolvimento do projeto tal como diminui os custos em recursos e treinamento de tecnologias específicas de cada plataforma alvo. Como desvantagens, podemos citar a limitação do uso do hardware do aparelho e a diminuição da performance do aplicativo em comparação à aplicações nativas em seus sistemas operacionais.

Como continuidade ao trabalho apresentado, a empresa responsável pelo aplicativo, a Associação de defesa dos consumidores Proteste, pretende realizar atualizações no aplicativo, transformando-o em uma plataforma colaborativa. Desta

forma, os próprios usuários poderão informar postos e preços, descentralizando o fornecimento da informação por parte da empresa. A atualização ainda contará com novas funcionalidades - como a possibilidade dos usuários poderem denunciar postos irregulares –, correções de bugs e reformulação da interface do aplicativo. Contudo, a nova versão continuará a utilizar o framework híbrido Phonegap.

Referências Bibliográficas

Rasmussen, B. (2013) “Os números não mentem: Android ou iOS, qual é o melhor?”, <<http://corporate.canaltech.com.br/analise/mobile/Os-numeros-nao-mentem-Android-ou-iOS-qual-e-o-melhor>>. Acessado em 2 de dezembro de 2014.

Moreira, E. (2012) “Nos principais mercados globais, já existem mais smartphones conectados que PCs”, <<http://www.techtudo.com.br/noticias/noticia/2012/01/nos-principais-mercados-globais-ja-existem-mais-smartphones-conectados-que-pcs.html>>. Acessado em 10 de dezembro de 2014.

Uol (2012) “Cinco anos do iPhone: veja números e curiosidades sobre o aparelho”, <<http://tecnologia.uol.com.br/infograficos/2012/06/28/5-anos-do-iphone-veja-numeros-e-curiosidades-sobre-o-aparelho.htm>>. Acessado em 10 de dezembro de 2014.

Iwata, M. (2013) “Comparativo entre as lojas de aplicativos Android, iOS e Windows Phone”, <<http://showmetech.band.uol.com.br/comparativo-lojas-aplicativos-android-ios-windows-phone>>. Acessado em 10 de dezembro de 2014.

Fasciana, C. (2014) “O que é PhoneGap?”, <<http://showmetech.band.uol.com.br/comparativo-lojas-aplicativos-android-ios-windows-phone>>. Acessado em 10 de dezembro de 2014.

Apache Software Foundation (2004) “Apache License”, <<http://phonegap.com/about/license>>. Acessado em 10 de dezembro de 2014.