

SSJFROTA: Um Aplicativo para Gerenciamento de Frota¹

Ubiratan Diego Cardozo da Rosa²

José Antônio de Figueiredo³

RESUMO

O uso das tecnologias aliado às necessidades do nosso dia a dia faz com que cresça em número significativo o desenvolvimento de aplicativos móveis. Devido à necessidade em um melhor controle no gerenciamento de frota de veículos, foi proposto e desenvolvido o presente trabalho, uma planilha digital para controle de retirada e devolução de veículos. Utilizando o framework Ionic, foi possível criar uma ferramenta versátil e atrativa para implementar tal controle. Foi necessária a criação de um banco de dados para armazenar os dados processados possibilitando assim a criação de um relatório conforme a necessidade de se obter os dados. Para efetuar a comunicação entre tecnologias diferentes foi necessária a criação de um Web Service, estrutura essa que possibilitou criar o cenário capaz viabilizar tal solução. Tecnologias como leitor QR Code, IMEI, triggers e outras funcionalidades, fez com que o projeto ganhasse em atratividade e ganho de tempo no manuseio e inserção dos dados necessários.

Palavras-chave: Ionic. Restful. QR Code. IMEI.

1 INTRODUÇÃO

Uma empresa de comércio de medicamentos, da cidade de Passo Fundo, disponibiliza a seus colaboradores uma frota própria de veículos para utilização em horário de serviço. Normalmente o veículo está na posse de apenas um usuário, que fica responsável por manter o mesmo em dia com a mecânica, higiene e cuidados pertinentes ao bem, incluindo responsabilidade com as multas geradas por infringir leis. Sempre que retira um veículo, o colaborador assina um termo onde se compromete com todos os pontos abordados anteriormente.

¹ Trabalho de Conclusão de Curso (TCC) apresentado ao Curso de Tecnologia em Sistemas para Internet do Instituto Federal Sul-rio-grandense, Câmpus Passo Fundo, como requisito parcial para a obtenção do título de Tecnólogo em Sistemas para Internet, na cidade de Passo Fundo, em 2018.

² Graduando do Curso Tecnologia em Sistemas para Internet – Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense – Passo Fundo / RS – ubiratan_dcr@hotmail.com.

³ Orientador Mestre Professor do Curso Tecnologia em Sistemas para Internet – Instituto Federal de Educação, Ciência e Tecnologia Sul-Rio-Grandense – Passo Fundo / RS – jose.figueiredo@passofundo.ifsul.edu.br.

Porém, alguns setores desta empresa compartilham estes veículos entre diversos funcionários e até mesmo com outros setores. Atualmente este controle se dá através de uma planilha impressa, que é preenchida na retirada e na devolução do veículo.

O controle manual do compartilhamento de veículos gera uma dificuldade para identificar quem estava utilizando o veículo em determinado dia e horário. Todavia, cada busca nos arquivos dispensa muito tempo de um funcionário e nem sempre os dados buscados são encontrados. Outro problema é que muitas vezes os dados da planilha não são corretamente preenchidos, dificultando assim encontrar o motorista responsável por determinado deslocamento.

Dado este problema, é proposto neste trabalho o desenvolvimento de uma ferramenta de apoio, para preenchimento e controle eletrônico do uso dos veículos. O APP foi desenvolvido para sistemas Android, onde sua utilização poderá se dar em *smartphones* e *tablets*, com uma interface amigável e atrativa para o usuário. Buscando tornar a ferramenta mais atrativa ao usuário final, procurou-se desenvolver um aplicativo prático, de modo que as informações necessárias para o preenchimento correto dos dados não passe somente pelo processo de interação do usuário.

O presente trabalho está organizado da seguinte forma, seção 2, Trabalhos Relacionados, onde é apresentado como resultado o estudo sobre ferramentas que propõem entregar a mesma funcionalidade do trabalho proposto. Seção 3, Desenvolvimento, onde é abordado toda a arquitetura, modelagem e desenvolvimento do projeto. Seção 4, Resultados, apresenta o resultado final do aplicativo, como estrutura do Banco de Dados, Web Service e telas do Aplicativo, também são apresentadas as telas do APP com suas funcionalidades juntamente com o resultado dos testes executados. Seção 5, Considerações Finais, onde são apresentadas as ferramentas e linguagens utilizadas no desenvolvimento, abordando algumas funcionalidades que poderão ser agregadas futuramente. Seção *Abstract*, versão inglês do resumo do trabalho. Seção Referências, traz todas às referencias de trabalhos e livros consultados para o desenvolvimento do projeto.

2 TRABALHOS RELACIONADOS

A busca por uma solução no controle de veículos é complexa, no mercado existem vários aplicativos que oferecem muitas funcionalidades para tal propósito, desde integração de consultas como: buscar veículos, busca de CNH, controle de rota, controle de manutenções e outros. Porém não foi encontrada uma solução que seja específica para resolver o problema proposto. Nesta seção é apresentado o estudo feito em alguns sistemas relacionados. Foram estudadas algumas destas ferramentas e os resultados serão apresentados a seguir:

2.1 UTILCAR GESTÃO DE CARRO, MULTA E CNH

O aplicativo Utilcar⁴ é um “Sistema de gestão veicular”, no qual os condutores e proprietários de veículos têm informações úteis na gestão do veículo. A seguir estão listadas as principais funcionalidades do APP:

- Consultas ao DETRAN: na última versão estão liberadas consultas somente nos estados de Santa Catarina, Rio Grande do Norte, Rio Grande do Sul, Mato Grosso, São Paulo, Paraná e Minas Gerais. Esta consulta permite acesso aos mesmos dados quando se efetua a consulta diretamente no site do DETRAN, como, consulta de veículo, informação do licenciamento, multas, recursos de infrações e outros.
- Consulta FIPE: onde se pode encontrar de acordo com informações lançadas na consulta de veículos, o valor médio do seu bem.
- Consulta de veículo suspeito: através da placa de um veículo, é efetuada uma busca em sistemas da polícia, os retornos são: situação legal ou veículo roubado.
- Emergência: localização de locais como hospitais, bombeiros, mecânicos, postos e guincho. Quando selecionado um dos itens, é retornada uma lista com os locais encontrados e informações como distância, telefone e a localização caso necessite se deslocar até lá.

⁴ Disponível em: <<http://www.dsdevelop.com.br/utilcar>> Acesso em: 20 de Abril de 2017

- Avisos automáticos: conforme prévia configuração, o aplicativo utiliza a base do DETRAN para retornar alguns alertas, como, vencimento de multas, IPVA, CNH e outros.

2.2 FROTA BACKSITE - GESTÃO DE CARROS, MOTORISTAS E USUÁRIOS

O aplicativo Frota Backsite⁵ é um sistema de gestão, com funcionamento em diversas plataformas, permitindo o acesso em diversos dispositivos como: *tablets*, celulares e computadores. Uma solução completa desenvolvida para atender a necessidade de controle das empresas que disponibilizam veículos aos seus funcionários. A seguir serão listadas as funcionalidades incluídas no APP:

- Controle em tempo real;
- Locais de interesse;
- Função de análise do modo de condução;
- Informações sobre excesso de velocidade;
- Criação e gerenciamento de rotas;
- Controle de combustível;
- Gerenciamento das manutenções dos veículos;

A empresa que adquirir a ferramenta poderá controlar sua frota através de centro de custo, podendo assim separá-los em cada departamento ou setor. Não tive maiores informações acerca do aplicativo, devido seu acesso ser liberado apenas para pagantes, as informações listadas foram retiradas site do fabricante.

2.3 CONSIDERAÇÕES

Dentre as duas soluções listadas, o aplicativo Frota foi o que mais se aproxima ao SSJFrota na questão de características, a ideia inicial do aplicativo resultante deste trabalho é o controle dos veículos mantendo um histórico da sua utilização. O APP Frota possui dois dados adicionais na retirada que se refere à coleta da informação de local da retirada e informação de longitude e latitude para a questão de rastreabilidade.

⁵ Disponível em: <<http://www.backsite.com.br/aplicativo-frota>> Acesso em: 20 de Abril de 2017

A proposta do aplicativo SSJFrota é simplificar a forma em que o usuário registra as operações de retirada e devolução de veículos, mantendo todas as informações pertinentes para a empresa manter o controle, com isso ambos os lados serão beneficiados com tal aplicação.

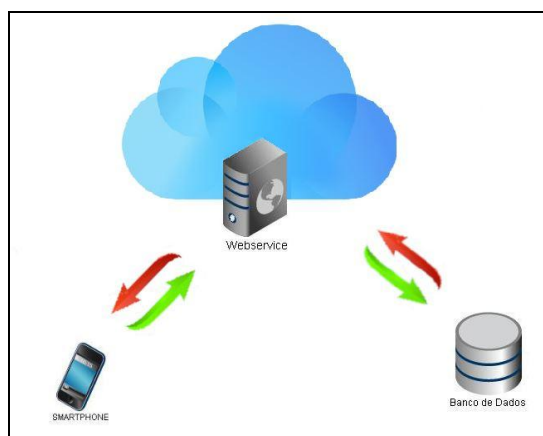
3 DESENVOLVIMENTO

Nesta seção é apresentado o planejamento do aplicativo, bem como dos processos necessários para viabilizar o gerenciamento da frota. Para o cenário proposto foi necessário à criação de um banco de dados, uma aplicação *Web Service* e o aplicativo Android.

3.1 ARQUITETURA DO SISTEMA PROPOSTO

O proposto trabalho foi idealizado para a resolução de problemas no controle da frota de veículo que são de uso em comum. O controle atual por ser em uma planilha impressa, por muitas vezes não se torna atrativo o seu preenchimento, o que se faz necessário para controle de quem estava de posse de um veículo em determina horário e local no caso de alguma infração de trânsito. O SSJFrota contempla basicamente as seguintes ações: retirada e devolução de veículos.

O desenvolvimento do aplicativo foi dividido em quatro etapas distintas, que são, modelagem do projeto, desenvolvimento do Banco de Dados, desenvolvimento do *Web Service* e desenvolvimento do aplicativo. Na figura 1 é possível verificar a arquitetura necessária para o funcionamento do sistema SSJFrota, sendo essas as fases necessárias no desenvolvimento da mesma.

Figura 1: Arquitetura do Projeto

Fonte: AUTOR, 2018

Tendo em vista que cada funcionário disponibiliza de um aparelho *smartphone* com um número corporativo para suas atribuições diárias, foi idealizada inicialmente a concepção de um aplicativo *mobile* com interface amigável e intuitiva.

Um dos fatores mais importante na realização de um projeto de programação é a usabilidade do sistema, pois o aplicativo pode ser atrativo ao usuário ou fazer do sistema uma ferramenta obsoleta. Segundo Laurel (1990) “Uma interface mal projetada pode causar danos graves e dificuldades na realização dos trabalhos para uma organização; mesmo assim, poucas organizações consideram esse fato quando estão planejando as suas aplicações” (apud Ferreira, 2003, p. 116). A questão que trata da interface foi fortemente vista no referido trabalho, tornar o aplicativo atrativo para o usuário é um grande desafio para os desenvolvedores.

Um banco de dados para receber as informações foi outro ponto analisado, pois a finalidade do aplicativo é obter as informações de maneira ágil, uniforme e confiável.

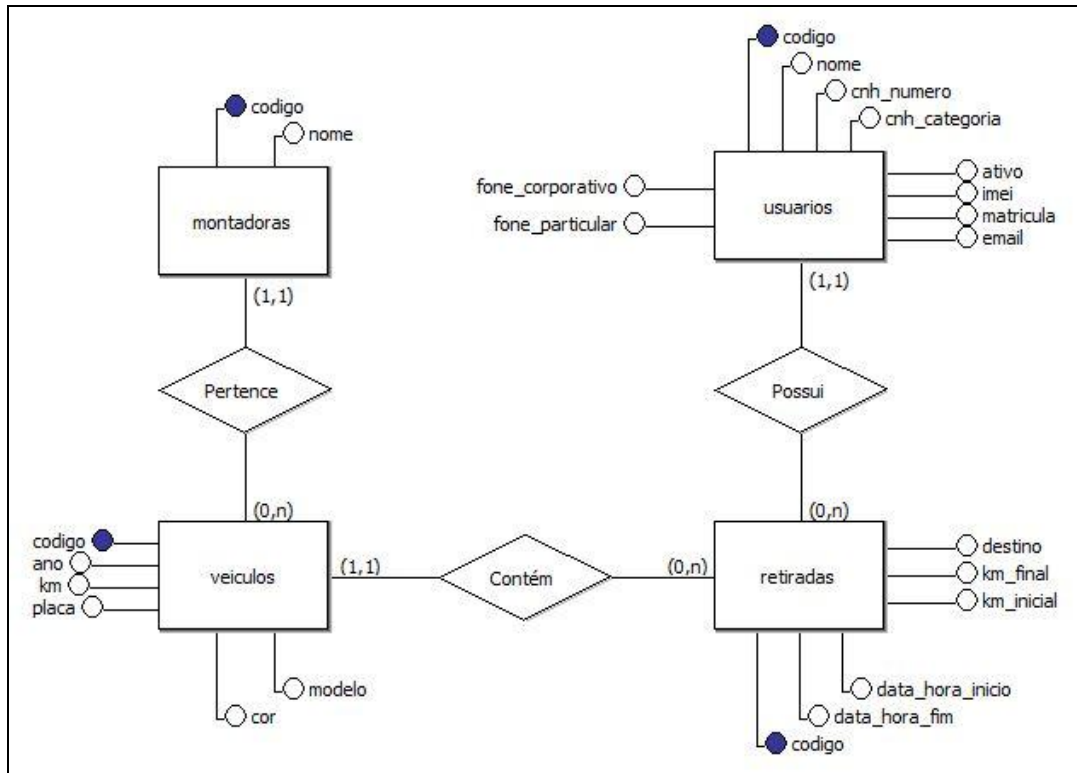
A aplicação Web Service entra no escopo de forma a ser a ferramenta de integração do aplicativo com o banco de dados.

3.2 MODELAGEM E DESENVOLVIMENTO DO BANCO DE DADOS

Por meio da ferramenta pgAdmin III foi desenvolvido o banco de dados com o nome SSJFrota, o mesmo é composto de quatro tabelas conforme modelo entidade

relacionamento apresentado na Figura 2. As tabelas planejadas são: montadoras, retiradas, usuários e veículos.

Figura 2: Modelo Entidade Relacionamento



Fonte: AUTOR, 2018

Para automatizar alguns processos foi necessária a criação de funções e triggers⁶ que contribuiriam de forma positiva para o resultado final, tirando assim a responsabilidade do usuário de ter que inserir todas as informações necessárias para o controle. A seguir veremos um detalhamento sobre a funcionalidade de cada função e trigger:

3.2.1 Funções

Função inserir_km_b_i(): Responsável inserir o valor no atributo **km_inicial** quando for inserido um registro de retirada no banco de dados, a função efetua um *update* no campo **km_inicial** da tabela **retiradas**, buscando na tabela **veiculos** a informação contida no atributo **km** do mesmo.

⁶ Gatilhos responsáveis por acionar as funções

Função `inserir_usuario_b_i()`: Responsável por buscar o código de usuário através do código IMEI⁷ inserido na retiradas, esta função compara o código inserido no atributo `imei_usuario` da tabela `retiradas`, com o código existente na tabela `usuarios` no campo `imei`, sempre restringindo a busca através do campo `ativo` da tabela `usuarios` que estiver assinalado como *true*.

Função `atualiza_km_a_u()`: Responsável por atualizar o atributo `km` na tabela `veiculos` toda vez que é efetuada uma devolução e informado o atributo `km_final` na tabela `retiradas`, ou seja, toda vez que houver um evento *update* no campo `km_final` da tabela `retiradas`, automaticamente será atualizado o atributo `km` da tabela `veiculos` com o valor inserido pelo usuário no ato da devolução.

Função `inserir_data_fim_b_u()`: Responsável por inserir data e hora atualizada no campo `data_hora_fim` sempre que houver o evento *update* nos atributos `destino` e `km_final`, indicando que houve uma devolução do veículo retirado.

3.2.2 Triggers

Os gatilhos foram todos vinculados à tabela `retiradas`, devido a maior parte das interações do aplicativo se dar com essa tabela, abaixo segue uma breve descrição.

Trigger `inserir_usuario_b_i`: Gatilho executado antes do evento *insert* na tabela `retiradas` é responsável por chamar a função `inserir_usuario_b_i()`.

Trigger `inserir_km_b_i`: Gatilho executado antes do evento *insert* na tabela `retiradas` é responsável por acionar a função `inserir_km_b_i()`.

Trigger `inserir_data_fim_b_u`: Gatilho executado após o evento *update* na tabela `retiradas`, é responsável por acionar a função `inserir_data_fim_b_u()`.

Trigger `inserir_km_b_i`: Gatilho executado após o evento *insert* na tabela `retiradas`, é responsável por acionar a função `inserir_km_b_i()`.

3.3 ESTRUTURA DO WEB SERVICE RESTFUL

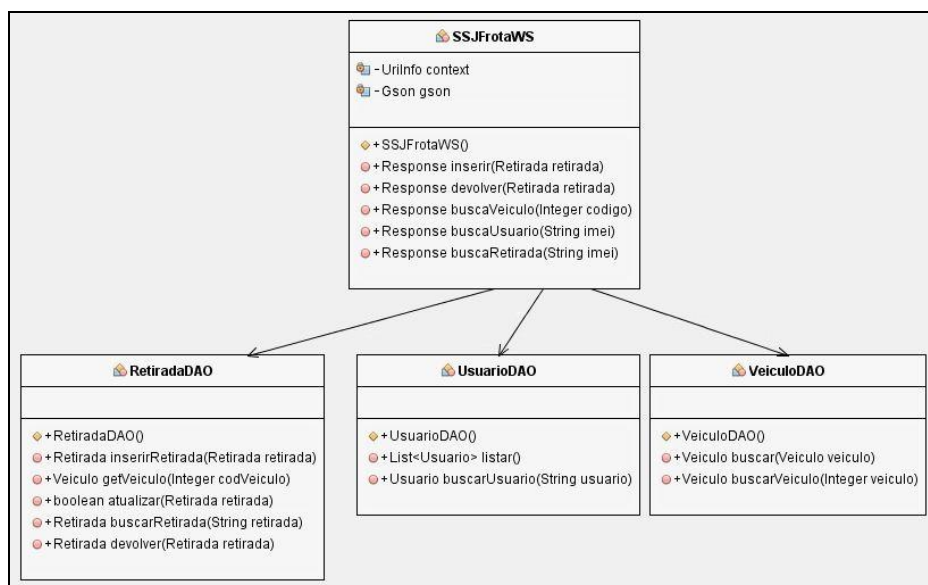
O Web Service foi codificado através da ferramenta NetBeans, construído com a arquitetura RestFul (Figura 3), em que o serviço faz a comunicação do aplicativo

⁷ IMEI - Identificação Internacional de Equipamento Móvel (*International Mobile Equipment Identity*).

com o banco de dados. Segundo Benatallah (2004), “A composição de Web Services tem surgido como uma importante estratégia para permitir a colaboração de aplicações entre empresas (Business-to-Business)” (apud SILVA, 2006, p. 1). A troca de mensagens se dá através de pacotes JSON, essa estrutura foi utilizada tendo em vista a facilidade e praticidade na hora de escrever e interpretar as informações. O serviço implementa o servidor de aplicação GlassFish⁸ 5.0. Foram utilizadas as requisições HTTP para transmissão de dados, são elas (Figura 3):

- @POST
 - inserir - para registrar uma retirada de veículo.
- @PUT
 - devolver - para registrar a devolução do veículo.
- @GET
 - buscaVeiculo/{codigo} - buscar dados do veículo através do seu código.
 - buscaUsuario/{imei} - buscar dados do usuário através do IMEI.
 - buscaRetirada/{imei} - busca se tem retirada em aberto com determinado IMEI passado como parâmetro.

Figura 3: Classes Web Service RestFul



Fonte: AUTOR, 2018

⁸ “É um servidor de aplicação *open source* liderado pela *Sun Microsystems* para a plataforma Java EE.” Disponível em: < <https://pt.wikipedia.org/wiki/GlassFish> > Acesso em: 27 de novembro de 2018

3.4 MODELAGEM E DESENVOLVIMENTO DO APLICATIVO ANDROID

O aplicativo por onde se dá toda a interação do usuário foi desenvolvido na ferramenta Visual Studio Code, interface essa que possibilitou a utilização da linguagem de programação *Type Script*⁹ através do framework Ionic¹⁰, cujo sua utilização permite compilar aplicações para os sistemas operacionais como: *Android*, *iOS* e *Windows Phone*, conforme podemos ver na Figura 4.

Figura 4: Ionic e suas Tecnologias



Fonte: PÁGINA DEVMIDIA, 2018¹¹

Conforme diagrama caso de uso mostrado na Figura 5, os processos implantados no aplicativo norteiam a retirada e a devolução do veículo, cada um deles aplicam suas funcionalidades de acordo com as necessidades de geração e coleta das informações. Tanto para retirar ou devolver um veículo existe um processo de atualização que será visto a seguir o seu funcionamento.

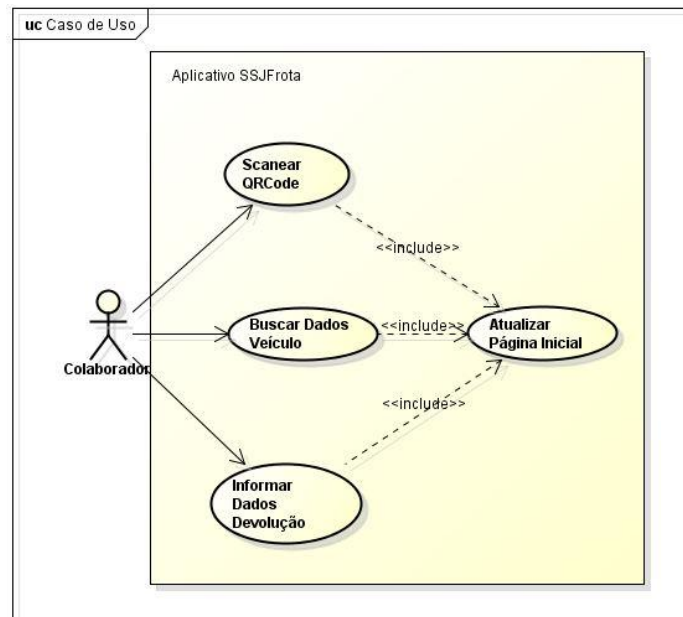
O diagrama de sequência (Figura 6) demonstra as funções implantadas para elucidar cada processo, deixando assim mais claro a lógica por trás do aplicativo.

⁹ TypeScript é uma linguagem derivada da linguagem *java script*, tipada, foi desenvolvida pela Microsoft.

¹⁰ O Ionic é um Kit de desenvolvimento de software completo e possibilita o desenvolvimento de aplicativos móveis.

¹¹ Disponível em: <<https://www.devmedia.com.br/guia/ionic/38372>> Acesso em: 17 de novembro de 2018

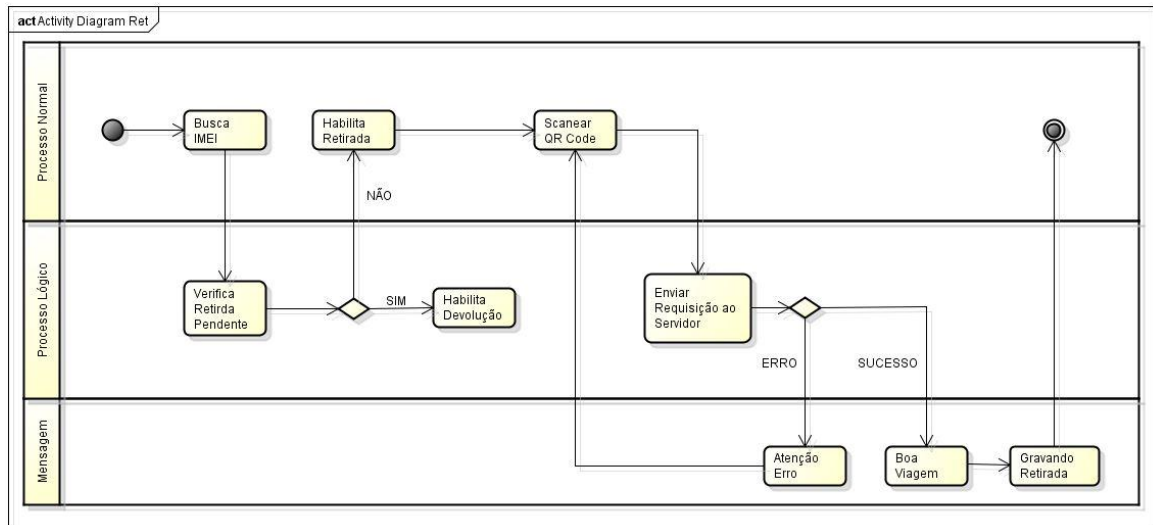
Figura 5: Diagrama Caso de Uso



Fonte: AUTOR, 2018

Assim, na Figura 6 é possível verificar como se dá o processo de retirada de um veículo, com os dados do usuário previamente carregados ao iniciar o aplicativo, a primeira verificação é feita na questão de existência ou não de uma retirada em aberto, não encontrando nenhum dado de retirada pendente, habilita-se o botão retirar que direciona para a respectiva tela. Nela encontramos um botão para escanear o código QR Code, que contém o código do veículo. Feito o processo citado, ao salvar os dados o sistema não identificando nenhum retorno de erro informa o sucesso na transação e grava os dados no banco de dados, finalizando assim o procedimento de retirada, caso encontre erro, é mostrado uma mensagem na tela informado o erro e o processo de retirada deve ser iniciado.

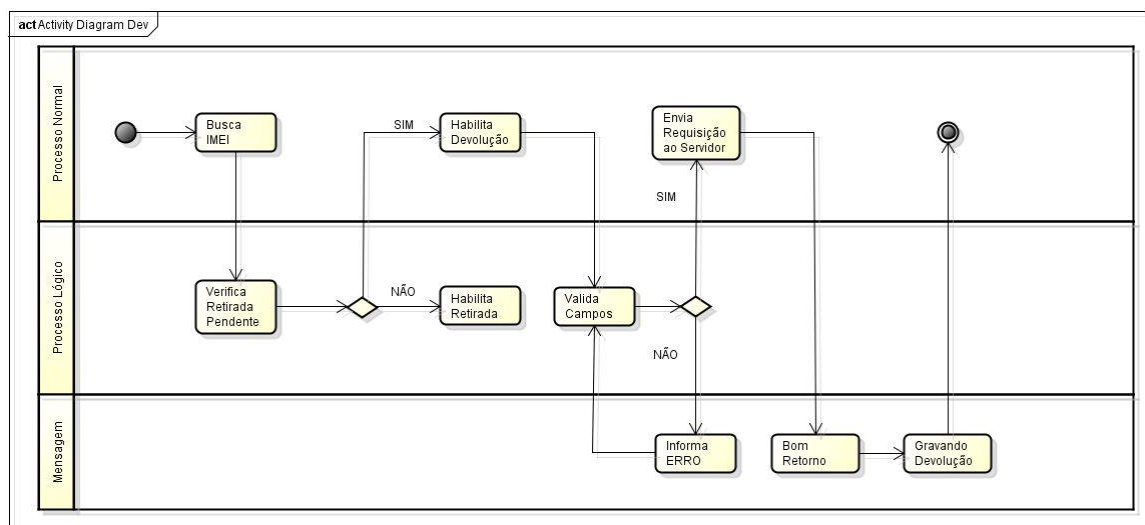
Figura 6: Diagrama de Sequencia de Retirada



Fonte: AUTOR, 2018

Já na Figura 7, podemos notar que o processo inicial é o mesmo, alterando quando é direcionado para a tela de devolução, na qual existe um processo de validação dos dados informados pelo usuário, que são o destino e a quilometragem final. A primeira validação é referente aos campos em branco, pois o sistema não aceita gravar os dados se não for informado nenhum dado nos dois campos, a outra validação é em relação ao KM, além de validar campo em branco, é feita uma validação de quilometragem, somente sendo aceito valor maior que a quilometragem inicial da retirada, que estará sempre disponível para verificação logo acima do campo de digitação.

Figura 7: Diagrama de Sequencia da Devolução



Fonte: AUTOR, 2018

4 RESULTADOS

A construção do aplicativo demandou vários processos até chegar ao resultado final. Para apresentar os resultados, serão utilizados os seguintes tópicos:

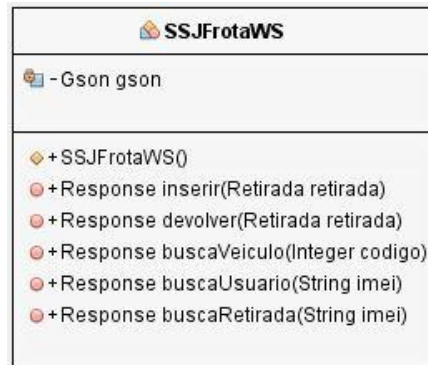
- Banco de dados
- Web Service
- Aplicativo e suas funcionalidades
- Testes

4.1 BANCO DE DADOS

A primeira etapa foi a modelagem e criação do banco de dados, em que foi utilizado o PostgreSQL, um sistema robusto de criação de banco de dados objeto-relacional. Foram mapeadas as informações pertinentes ao funcionamento do APP de onde surgiram quatro tabelas para armazenamento das informações. O uso de funções e triggers se fez necessário devido à necessidade de automatizar a inserção de alguns dados importantes, como, quilometragem inicial da retirada e atualização do campo km na tabela veículos e outros.

4.2 WEB SERVICE

Com o processo de criação do banco finalizado, o próximo ponto foi o desenvolvimento do Web Service com arquitetura RESTful, aplicação necessária por intermediar a comunicação do aplicativo com o banco de dados. A classe SSJFrotaWS foi implementada de modo que as mensagens de *request* e *response* sejam realizadas através do formato JSON, conforme ilustrado na Figura 8.

Figura 8: Classe SSJFrotaWS

Fonte: AUTOR, 2018

4.3 APLICATIVO E SUAS FUNCIONALIDADES

O aplicativo teve como resultado três telas intuitivas e com poucas informações a serem inseridas pelo usuário, visando agregar facilidade na sua utilização.

A construção das telas foi pensada de modo que todas as informações pertinentes aos processos de retirada e devolução de veículo ficassem disponíveis ao usuário.

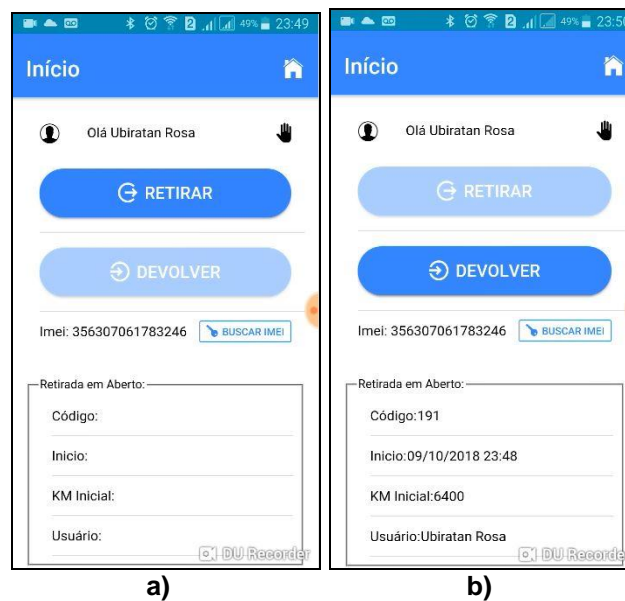
A tela inicial (Figuras 9.a e 9.b) possui três áreas em seu corpo, a primeira traz uma saudação com o nome do usuário, para buscar essa informação é disparado uma requisição GET com o código IMEI do aparelho, no qual a busca é feita na tabela usuários do banco de dados, comparando o código IMEI enviado com a informação já cadastrada, retornando apenas por usuário que esteja marcado como ativo.

A segunda área é composta por botões, “RETIRAR”, “DEVOLVER” e “BUSCAR IMEI”, os dois primeiros são responsáveis por direcionar para as respectivas telas de retirada e devolução, os mesmos foram codificados de modo a ficar apenas um deles ativo, para isso, foi criada uma rotina que busca informação de retirada em aberto no banco de dados e caso não tenha nenhum dado para retornar habilita o botão “retirar” conforme Figura 9.a, Caso contrário, encontrando dados de retirada em aberto, desabilita o botão “retirar”, ficando ativo apenas o botão “devolver” conforme Figura 9.b, este processo foi criado para evitar que um usuário tenha dois veículos vinculados a ele sem a devida devolução. O botão “buscar IMEI” foi incluído para caso ocorra alguma anormalidade na busca automática pelo APP, assim o

usuário tem esta alternativa para dar sequência no processo de retirada ou devolução.

A última área da tela inicial é o campo “retirada em aberto”, caso a rotina que busca informação de retirada em aberto retornar dados, Figura 9.b, conforme já mencionado, o botão devolver é ativado e o campo é preenchido com as devidas informações da retirada em aberto.

Figura 9: Tela inicial

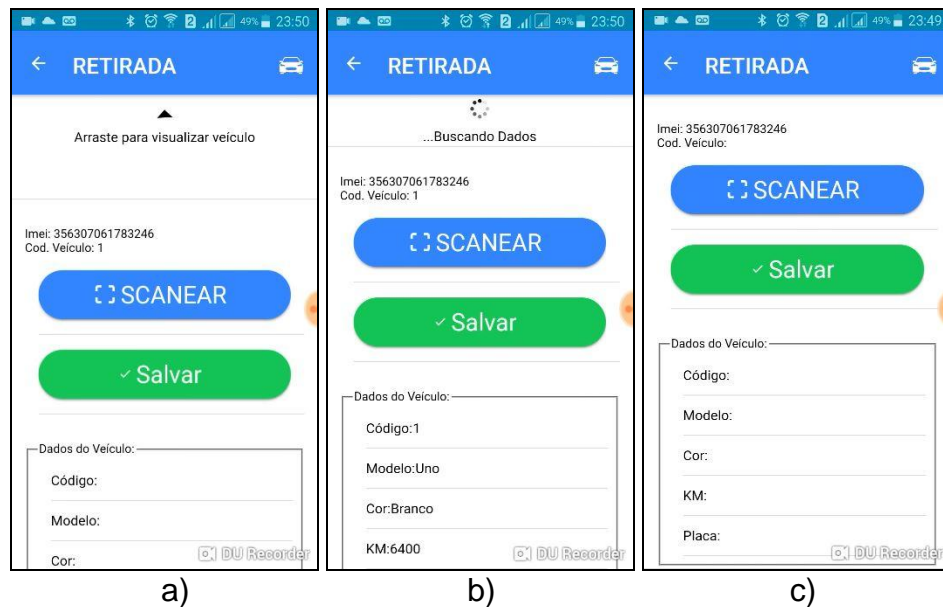


a) Fonte: AUTOR, 2018 b)

Ao clicar no botão retirar, Figura 9.a, o usuário é direcionado para a segunda tela (Figura 10.c), onde será feito o processo de retirada de um veículo.

A retirada passa a ser um processo muito simples, inicialmente o usuário precisa escanear um código QR Code que estará colocado estrategicamente no veículo, em seu painel, feito este passo, existe um campo logo abaixo do botão salvar, onde com um processo de atualização de página conforme Figura 10.a, serão informados os dados do carro Figura 10.b, são eles: código, modelo, cor, km_inicial e placa, após a visualização e conferência das informações, ao clicar no botão “devolver” a tela é direcionada para o início e a retirada é gravada no banco.

Figura 10: Tela Retirada

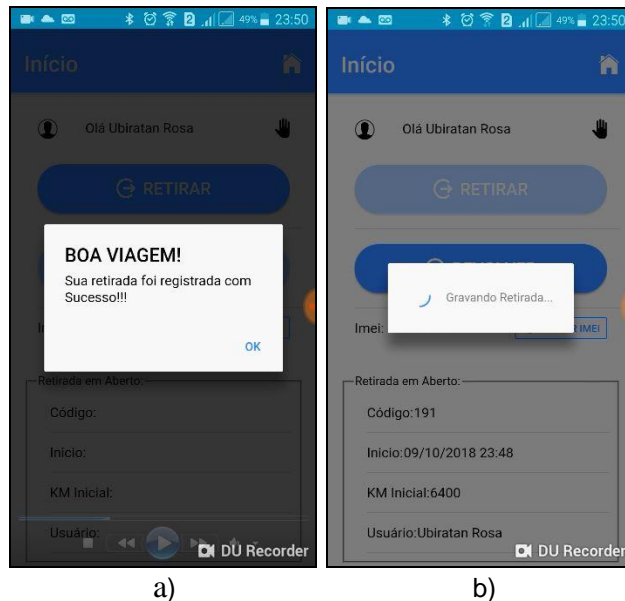


Fonte: AUTOR, 2018

Após salvar os dados, é mostrado um alerta na tela, desejando-lhe boa viagem (Figura 11.a), posteriormente roda um processo de carregamento *loading*¹² (Figura 11.b), durante um período de 20 segundos até que a página seja carregada com as informações da retirada (Figura 11.b), desabilitando assim o botão “retirar” e habilitando o botão “devolver” (Figura 9.b).

¹² “O componente *Loading* é uma sobreposição que impede a interação do usuário ao indicar atividade.” (IONICFRAMEWORK, 2018)

Figura 11: Informações da Retirada



Fonte: AUTOR, 2018

Para efetuar uma devolução, o usuário será direcionado para a tela de devolução (Figura 12), onde será necessário informar basicamente dois dados, o primeiro campo é o destino que deu origem a retirada e o segundo a quilometragem final da devolução.

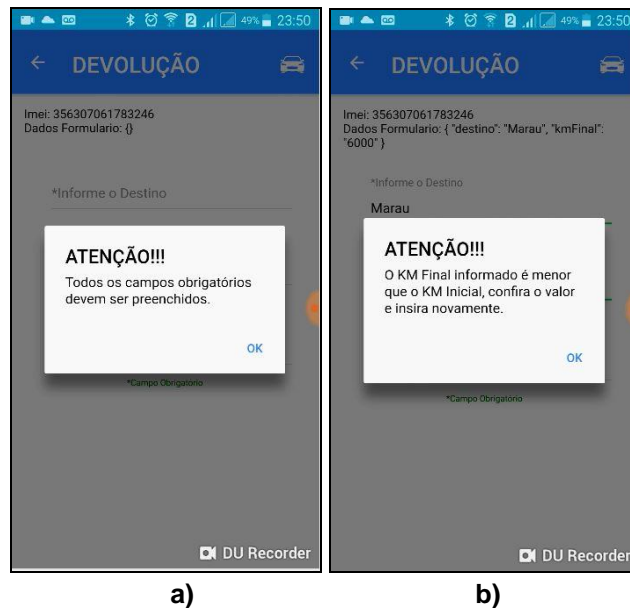
Figura 12: Tela Devolução

Fonte: AUTOR, 2018

Conforme Figuras 13.a e 13.b, os dois campos estão configurados para validar o seu preenchimento, ou seja, o campo devolução não pode ser enviado em branco, já

o campo km além da verificação de preenchimento, é feita a validação do valor informado, o mesmo não pode ser menor e nem igual ao valor inicial que foi carregado diretamente do banco de dados e informado na tela inicial (Figura 9.B) e na tela de devolução (Figura 12).

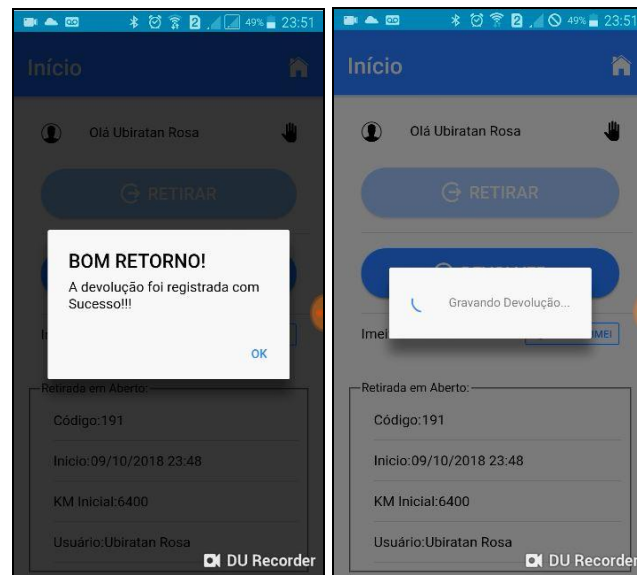
Figura 13: Validação da Devolução



Fonte: AUTOR, 2018

Após a inserção correta dos dados, uma mensagem é mostrada informando o sucesso ao gravar os dados (Figura 14.a), o usuário é direcionado novamente para a tela inicial ficando apto a retirar outro veículo, após a rotina de atualização (Figura 14.b).

Figura 14: Mensagens de Devolução



a) b)

Fonte: AUTOR, 2018

4.4 TESTES

Os testes se deram no cenário de desenvolvimento, para verificação e aferição da programação implantada, tendo como um totalizador de 200 inserções no banco de dados, especificamente na tabela retiradas, englobando todas as etapas do desenvolvimento. Foram ratificadas as funções e *triggers* implantadas diretamente no banco de dados, verificado o pleno funcionamento das mesmas. Da mesma forma, os processos do aplicativo que utilizam da intermediação do Web Service para os processos de envio e recebimento dos dados, tiveram os seus funcionamentos aferidos e passaram nos testes.

Foram utilizados ao todo três aparelhos com plataforma Android, para simular os usuários, com a inserção manual dos dados na tabela usuários. Inclusive, para verificar a questão do usuário ativo na tabela, foi feito o cadastro de dois usuários com o mesmo código IMEI, deixando um com o atributo ativo como *true* e o outro ativo como *false* e alternando essa informação entre eles, em todos os testes apenas o usuário ativo tinha seus dados vinculados à retirada conforme era esperado.

5 CONSIDERAÇÕES FINAIS

O presente trabalho nos mostra como resultado uma ferramenta prática e de fácil utilização, além de ser atrativa para o usuário, é focada em resolver a real necessidade de controle interno da empresa.

Esse trabalho nos apresenta não apenas um APP e sim um ambiente completo para gerir veículos. Foi necessária a criação de um Banco de Dados com linguagem SQL para receber as informações, a solução foi desenvolvida através da ferramenta de SGBD¹³ pgAdmin III. Também foi construído um aplicativo para plataforma Android, utilizado para coletar do colaborador as informações necessárias dentro do processo de retirar e devolver um veículo, em que sua concepção se deu através do programa Visual Studio Code e utilizou-se da linguagem *Type Script* e diversos outros componentes para tornar a ferramenta usual e atrativa. Para comunicação entre as linguagens diferentes foi projetada e construída a aplicação *Web Service*, utilizando-se da IDE NetBeans e linguagem Java Script, essa possibilitou a conexão e interpretação dos dados em ambos os lados.

Durante o trabalho verificou-se diversas funcionalidades que futuramente podem ser agregadas ao projeto, como, tecnologia de geolocalização para rastreamento de rotas, até mesmo a utilização da câmera do celular para capturar informações do painel como os dados do odômetro liberando o usuário da necessidade de digitar textos durante o processo de devolução.

Além disso, verificou-se a necessidade de criação de um ambiente web para gerir estas informações, pois quando houver a necessidade de inserção de usuários, veículos e montadoras, é necessário realizar o procedimento diretamente no banco de dados. Além disso, também se faz necessário à implementação de relatórios e filtros de busca para a gestão dos dados neste ambiente.

O resultado de todo este trabalho é uma ferramenta cujo sua aplicação é específica para determinada empresa, mas se encaixa perfeitamente para outras empresas que disponibilizam de veículo próprio para funcionários efetuarem suas atividades diárias.

¹³ Sistema de Gerenciamento de Banco de Dados.

ABSTRACT

The use of technologies in addition to the needs of our day-to-day operations means that mobile applications are growing significantly. In addition to the need for a better control in vehicle fleet management, the present work was proposed and developed, a digital spreadsheet for vehicle recall and return control. Using the Ionic framework, it was possible to create a versatile and attractive tool to implement such a control. It was necessary to create a database to store the processed data, thus enabling the creation of a report according to the need to obtain the data. In order to communicate between different technologies, it was necessary to create a Web Service Restful, a structure that made it possible to create the scenario capable of enabling such a solution. Technologies such as QR Code reader, IMEI, triggers and other functionalities, made the project gain in attractiveness and time gain in the handling and insertion of the necessary data.

Keywords: Ionic. Restful. QR Code. IMEI.

REFERÊNCIAS

CRUZ, Vitor Silva; PRETUCELLI, Erick Eduardo. TECNOLOGIAS WEB PARA O DESENVOLVIMENTO MOBILE NATIVO. 2017. Disponível em: <<http://simtec.fatectq.edu.br/index.php/simtec/article/view/269>>. Acesso em: 17 de outubro de 2018.

DA SILVA, Arnaldo N. et al. Avaliação de desempenho da composição de web services usando redes de petri. In: Brazilian Symposium on Computer Networks. Curitiba, Paraná, Brazil. 2006. Disponível em: <http://repositorio.ufc.br/bitstream/riufc/24482/1/2006_eve_ansilva.pdf>. Acesso em: 23 de maio de 2017.

FERREIRA, S. B. L.; LEITE, J. C. S. P. Avaliação da usabilidade em sistemas de informação: o caso do Sistema Submarino. Revista de Administração Contemporânea, v. 7, n. 2, p. 115-136, 2003. Disponível em: <http://www.scielo.br/scielo.php?pid=S1415-65552003000200007&script=sci_arttext&tlng=pt>. Acesso em: 03 de junho de 2017.

SITE IONICFRAMEWORK. Componentes Ionic. Disponível em: <<https://ionicframework.com/docs/components/>> Acesso em: 15 de outubro de 2018