

**CENTRO PAULA SOUZA
FACULDADE DE TECNOLOGIA DE FRANCA
“Dr. THOMAZ NOVELINO”**

TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

EDUARDO PUGLIESI ASSIS LIMA

SISTEMA PARA GESTÃO DE MANUTENÇÃO DE VEÍCULOS

Trabalho de Graduação apresentado à Faculdade de Tecnologia de Franca - “Dr. Thomaz Novelino”, como parte dos requisitos obrigatórios para obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Me. Antônio Clementino Neto

FRANCA/SP

2022

SISTEMA PARA GESTÃO DE MANUTENÇÃO DE VEÍCULOS

Eduardo Pugliesi Assis Lima¹

Resumo

Este trabalho de graduação propõe o desenvolvimento de um software de manutenção de frota de veículos que deverá, por meio de entrada de dados realizada pelo usuário de maneira rápida e simples, armazenar os registros de manutenções dos veículos e através de cálculos com o parâmetros fornecidos pelo gestor, alertar para futuras manutenções. Pelas características da empresa alvo do produto, o sistema priorizou o funcionamento em arquitetura *desktop* na estação de trabalho do operador, afim de poupar recursos de processamento gastos por um navegador web, que podem ultrapassar 1 Gigabyte de memória RAM. O objetivo do sistema é auxiliar as empresas de transportes a terem maior controle das condições de operação dos veículos da frota. Foi utilizada a linguagem de programação Python para implementar as funcionalidades e a biblioteca Tkinter para o desenvolvimento da interface gráfica do sistema. Os dados foram persistidos em um banco de dados SQLite pela facilidade de implementação e economia de recursos da máquina. As tecnologias usadas são dependências padrões do Python3. A metodologia de análise, projeto, implementação e testes usada no projeto foram as apresentadas e praticadas durante as aulas de Engenharia de Software do curso de ADS (Análise e desenvolvimento de sistemas) da Fatec Franca.

Palavras-chave: Desenvolvimento de software. Gestão de frotas de veículos. Manutenção de veículos. Python. Tkinter.

Abstract

This paper aims to develop a system to control vehicle maintenance. The developed system aims to help transport companies to have greater control over the operating conditions of fleet vehicles. The software must, through data entry performed by the user in a quick and simple way, store the vehicle maintenance records and, through calculations with the parameters provided by the manager, alert for future maintenance. Due to the characteristics of the target company of the product, the system prioritized the operation in desktop architecture on the operator's workstation, in order to save the processing resources spent by a web browser, which can exceed 1 Gigabyte of RAM memory. That means almost 99% savings compared to the prototype developed for this system. The Python programming language was used to implement the features and the Tkinter library for the development of the graphical interface of the system. The data was persisted in a SQLite database for ease of implementation and economy of machine resources and the technologies used are standard Python3 dependencies. The analysis, design, implementation and testing

¹Graduando em Análise e Desenvolvimento de Sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: [pugliesiedu@gmail.com].

methodology used in the project were those presented and practiced during the Software Engineering classes of the Fatec Franca SA (System Analysis) course.

Keywords: Python. Software development. TKInter. Vehicle fleet management. Vehicle maintenance.

1 Introdução

Com o crescente aumento da tecnologia digital de informação e comunicação (TDIC), ficou cada vez mais simples automatizar processos de operacionais e aumentar a eficiência da gestão no mundo corporativo. De forma segura e confiável, o mundo passou a presenciar cada vez mais os softwares substituindo as velhas e antigas cadernetas de papel, o que trouxe cada vez mais agilidade e eficiência para as empresas.

Em empresas transportadoras, sejam elas pequenas, médias ou grandes, possuem veículos de diversos tipos, para transportes de pessoas, equipamentos ou cargas. Dentre esses veículos, pode-se destacar caminhões de grande e pequeno portes, vans e carros. Grande parte dessas empresas não possuem um controle preciso das condições de operação e das manutenções de sua frota.

Tendo proximidade com proprietários de empresas do ramo de transportes, foi percebido que há dificuldade na gestão das manutenções preventivas de seus veículos, tendo sido relatado por estes empresários, haver casos em que uma simples troca de óleo não foi realizada por falta de controle. Segundo matéria publicada pelo jornal Gazeta do Povo (TROVÃO, 2018), uma retífica de motor em veículos pequenos, gera despesas que começam em R\$ 3.500,00 e podem ir até R\$ 8.000,00. No mundo dos caminhões, os custos sobem absurdamente, onde apenas o cabeçote de um motor de caminhão, segundo dados levantados na loja oficial da Volvo (VOLVO, *sd*), custam a partir de R\$ 26.000,00.

Independente do tamanho da empresa, despesas dessa proporção, quando não planejadas, acarretam dificuldade na gestão do caixa ou até mesmo endividamentos. A melhor forma de se evitar despesas desnecessárias com veículos, é realizando as manutenções preventivas.

Com base nisso, ficou claro o problema de projeto a ser resolvido: projetar e implementar um sistema que seja rápido, fácil e simples de operar para facilitar a

gestão da manutenção de veículos.

O sistema, por meio de uma tela simples e intuitiva, terá a função de registrar os dados de um veículo, seja ele de pequeno, médio ou grande porte. Após o registro do veículo, o sistema fornecerá ferramentas para possibilitar o gestor inserir manutenções realizadas e agendar manutenções preventivas. Através de um campo responsável pela coleta da quilometragem atual do veículo (que será de preenchimento obrigatório), o sistema deverá realizar uma comparação e estimar se a “quilometragem atual” inserida pelo usuário é equivalente ou ultrapassa alguma quilometragem agendada para realização de alguma manutenção programada. Caso o resultado dessa comparação seja que, sim o veículo atingiu aquela quilometragem alvo, o sistema produzirá um alerta visual para avisar o usuário do sistema.

2 Viabilidade do projeto

Para analisar a viabilidade do projeto, foi utilizada a ferramenta Business Model Canvas (BMC) que permite de um forma gráfica e resumida, entender os principais pontos que impactam o sucesso de um empreendimento.

A Figura 1 apresenta as características do negócio em cada um dos 9 blocos da metodologia proposta por Alexander Osterwalder (ABSTARTUPS, 2019)

Os blocos da direita (Relacionamento, Segmentos de clientes, e Canais) se referem às características dos clientes que usam o aplicativo; o bloco central (Oferta de valor) caracteriza o produto a ser oferecido para resolver as necessidades dos clientes; os da esquerda (Parcerias-chave, Atividades-chave, e Recursos-chave) se referem às características operacionais do negócio; e os inferiores (Estrutura de custos e Fontes de receita) aos processos econômico-financeiros para sustentação do negócio.

A sequência sugerida de definição dos 9 blocos é: 1. Segmento de Clientes; 2. Proposta de Valor; 3. Canais; 4. Relacionamento com Clientes; 5. Fontes de Receitas; 6. Recursos Principais; 7. Atividades Principais; 8. Parcerias Principais; e 9. Estrutura de Custos. Com este método de análise e avaliação do produto, têm-se uma visão mais clara de como desenvolvê-lo de forma que entregue valor ao cliente e seja sustentável ao longo do seu ciclo de vida.

3 Levantamento de Requisitos

Nesta seção são apresentados os artefatos de documentação, análise e projeto do sistema, como preconizado pela Engenharia de Software.

3.1 Elicitação e especificação dos Requisitos

O levantamento de requisitos foi realizado através de uma entrevista semiestruturada com o proprietário de uma transportadora e com diversos veículos. Durante o encontro, foi relatada a dificuldade de se gerenciar as condições, comumente mecânicas, dos veículos de sua transportadora, especialmente aqueles veículos que são de uso coletivo pelos funcionários da empresa. Durante a entrevista, era visível a frustração do empresário, pois nestes veículos de uso coletivo, os funcionários sempre deixavam para os outros colegas a responsabilidade de fazer uma simples conferência da validade do óleo do motor, chegando ao ponto do carro passar dos 5000 quilômetros de atraso para a troca de óleo do motor.

Outra reclamação foi que, quando alguém fazia a manutenção de um veículo, a comunicação à gerência muitas vezes não era feita, ou quando era, nenhum registro era realizado, o que acabava gerando a dificuldade de controle das condições reais de operação do veículo.

Como resultado do levantamento, foi solicitado um sistema no qual, de forma simples, rápida e intuitiva, fosse possível cadastrar veículos e cadastrar manutenções destes veículos para consultas e acompanhamento da situação da frota.

Figura 1 – BMC do projeto

Parcerias-chave - Empresas de publicidade - Patrocinadores	Atividades-chave - Gerenciar, armazenar e planejar manutenções de veículos	Oferta de valor - Produto gratuito; - Simplicidade de uso; - Simplicidade para começar a usar;	Relacionamento - Atendimento direto por chat exclusivo de parceiros ou redes sociais	Segmentos de clientes - Transportadoras ou empresas com grande frota de veículos.
	Recursos-chave - Desenvolvedores - Equipe de suporte		Canais - Site Oficial - Redes Sociais	
Estrutura de custos - Mão de Obra qualificada			Fontes de receita - Doações e patrocínios	

Fonte: o autor

Sob o aspecto de usabilidade, foi solicitado pelo cliente que a interface deveria ter botões, textos e campos para inserção de dados com grandes dimensões, para eventuais usuários com algum tipo de dificuldade visual, garantindo a acessibilidade.

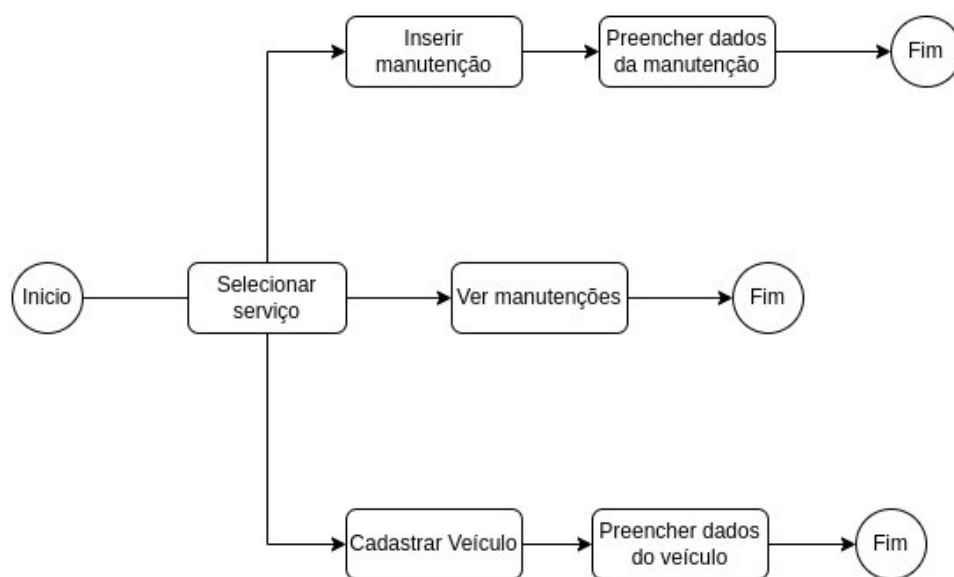
Em relação à segurança dos dados, foi requerido que fosse simplificado o processo de *backups* preventivos para eventuais recuperações de dados caso problemas técnicos ocorressem nos computadores.

O sistema deveria ser leve e capaz de rodar em computadores de baixo desempenho e qualidade de hardware, tendo sido descartada a implementação por uma aplicação WEB, visto que os navegadores atuais consomem pelo 1 Gigabyte de memória RAM, enquanto o primeiro protótipo compilado desse sistema, apenas 14 Megabytes. O sistema deveria ser compatível com o os sistemas operacionais Windows e Linux, pois também está sendo analisada a possibilidade de migração dos sistemas operacionais usados nos computadores da empresa, das atuais versões Windows para distribuições Linux mais leves, por razões econômicas.

3.2 Business Process Model and Notation (BPMN)

A Figura 2 mostra o diagrama BPMN do sistema projetado, com os principais processos automatizados.

Figura 2 – Diagrama BPMN dos processos



Fonte: o autor

3.3 Requisitos Funcionais

Os Requisitos Funcionais levantados na fase de análise estão descritos no Quadro 1.

Quadro 1 – Requisitos Funcionais do sistema

RF001 – Cadastrar veículos	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve cadastrar e exibir uma lista de veículos.		
RF002 – Registrar manutenções realizadas nos veículos cadastrados.	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve ser capaz de registrar manutenções dos veículos cadastrados.		
RF003 – Programar manutenções	Categoria:	Prioridade:

	() Oculto (X) Evidente	(X) Altíssima () Alta () Média () Baixa
Descrição: O sistema deve ser capaz de agendar manutenções, por quilometragem ou data.		
RF004 – Alertar usuário sobre manutenções agendadas	Categoria: () Oculto (X) Evidente	Prioridade: (X) Altíssima () Alta () Média () Baixa
Descrição: Com base nos dados inseridos pelo usuário, o sistema deve alertar futuras manutenções.		

Fonte: o autor

3.4 Requisitos Não Funcionais

Os Requisitos Não Funcionais que caracterizam o projeto estão apresentados no Quadro 2.

Quadro 2 – Requisitos Não Funcionais do sistema

RNF001	– O sistema deve ser leve e rápido.	Tipo	() Desejável (X) Obrigatório	(X) Permanente () Transitório
RNF002	– O sistema deve rodar em Windows e Linux.	Tipo	(X) Desejável () Obrigatório	(X) Permanente () Transitório
RNF003	– O sistema deve armazenar data de inserção e manutenções dos veículos	Tipo	() Desejável (x) Obrigatório	(X) Permanente () Transitório
RNF004	– O sistema deve armazenar data de inserção e manutenções dos veículos	Tipo	() Desejável (x) Obrigatório	(X) Permanente () Transitório

Fonte: o autor

3.5 Regras de Negócio

As Regras de Negócios identificadas no levantamento de requisitos e análise do problema estão descritas no Quadro 3.

Quadro 3 – Regras de Negócio do sistema

RN001 - Cadastros
Descrição: Será permitido o cadastro de qualquer tipo de veículo, desde que tenha Placa, Marca, Modelo e ano declarados.
RN002 - Atualizações

Descrição: O sistema deverá registrar qualquer tipo de atualização, desde a uma atualização mecânica até um rodízio de pneus.
RN003 - Login
Descrição: O acesso ao sistema será exclusivo a pessoas ligadas diretamente a administração da empresa, restringindo através de Login e Senha dos responsáveis.
RN004 – Requerimentos de dados para cadastro de veículos
Descrição: Será obrigatório o preenchimento de todos os campos disponíveis para efetuar o cadastro de um veículo
RN005 – Requerimentos de dados para registro de manutenções
Descrição: Será obrigatório o preenchimento apenas dos campos “Manutenção” e “KM Atual” na janela de atualização de veículos.

Fonte: o autor

3.6 Casos de Uso

Os Casos de Uso identificados na fase de análise do sistema, são mostrados no diagrama da Figura 3.

3.7 Diagrama Entidade-Relacionamento (DER)

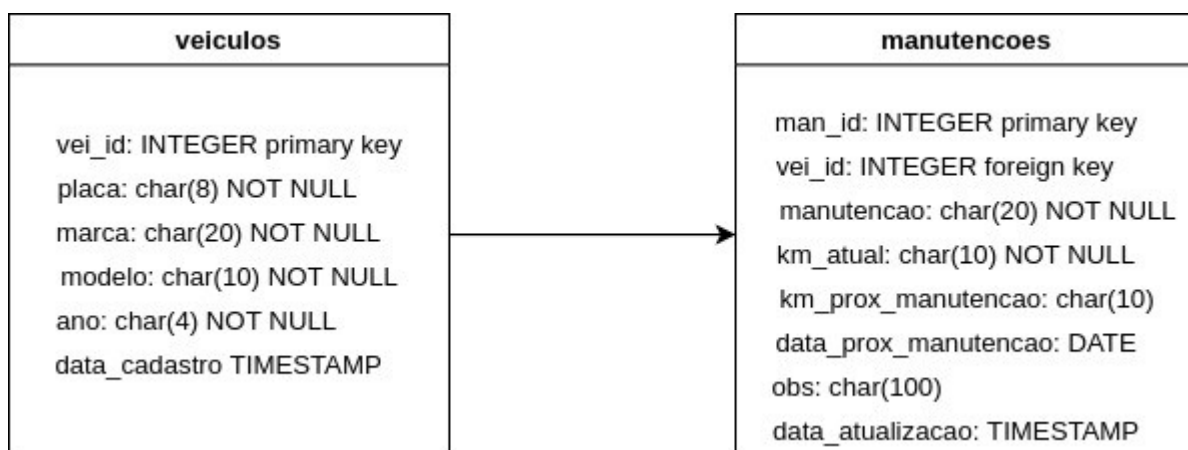
Para a persistência dos dados no banco, as tabelas foram modeladas e o diagrama é apresentado na Figura 4.

Figura 3 – Diagrama de Casos de Uso do sistema



Fonte: o autor

Figura 4 – Diagrama Entidade-Relacionamento do banco de dados



Fonte: o autor

4 Ferramentas e Métodos ou Desenvolvimento

4.1 Ferramentas

Para desenvolvimento do sistema, foi utilizada a linguagem de programação Python, na versão 3.10. Para a implementação da interface gráfica de operação foi utilizada a biblioteca Tkinter, que é a mais utilizada no desenvolvimento com Python.

Para implementação do banco de dados foi utilizado o SQLite3, em sua versão 3.34, e para visualização das tabelas, foi usado o software gratuito DB Browser for SQLite, na sua versão 3.11.2.

Para compilação do código Python foi usada a ferramenta Pyinstaller em sua versão 5.0, e para o desenvolvimento do código foi utilizada a IDE VSCode na versão 1.66.

A máquina utilizada para o projeto está com o sistema operacional Zorin OS na versão 16, que tem como base a LTS 20.04 do sistema operacional Ubuntu.

Para criar os diagramas, foi utilizado o aplicativo web gratuito Draw.io, e para redação deste documento foi utilizado o sistema de código aberto Libre Office versão 7.3.

4.2 Métodos ou Desenvolvimento

O desenvolvimento do projeto foi um processo significativamente desafiador. Para o desenvolvimento das janelas, a dúvida inicial foi de como elas iriam se comportar, e obviamente, seguir os requisitos solicitados pelo cliente de manter textos, campos a serem preenchidos e botões grandes, para facilitar a visualização. O Tkinter, ferramenta utilizada para fazer a interface gráfica, fornece alguns recursos para o comportamento das janelas, entre eles a função de travar uma janela que fique atrás da janela aberta (em foco) para inserção de dados, fazendo com que a única janela acessível para o usuário seja a que está de forma primária na tela.

Ainda em relação à interface, outras pequenas dificuldades que logo foram superadas, se referem à montagem das *listboxes* para exibição da lista de veículos cadastrados e manutenções a serem realizadas.

Em relação à lógica de operação do sistema e a implementação das suas regras de negócio o foco maior foi a montagem das tabelas do banco de dados, onde inicialmente a rotina verifica se já existe um banco de dados no diretório padrão antes de realizar a montagem dele de fato, para acesso às tabelas de dados.

As conexões das duas tabelas do banco, são feitas de modo separado em duas janelas de operação diferentes, já que uma tabela remete aos veículos e outra tabela aos registros das manutenções realizadas nos veículos cadastrados.

5 Resultados e Discussão

Através da metodologia e das ferramentas utilizadas, foi possível cumprir a maior parte dos objetivos iniciais. Entre eles destaca-se: modelagem inicial de um banco de dados, conexão do *script* Python com o banco de dados, manipulação completa dos dados inseridos no banco de dados, todas as janelas funcionais e compilação de um protótipo funcional do sistema.

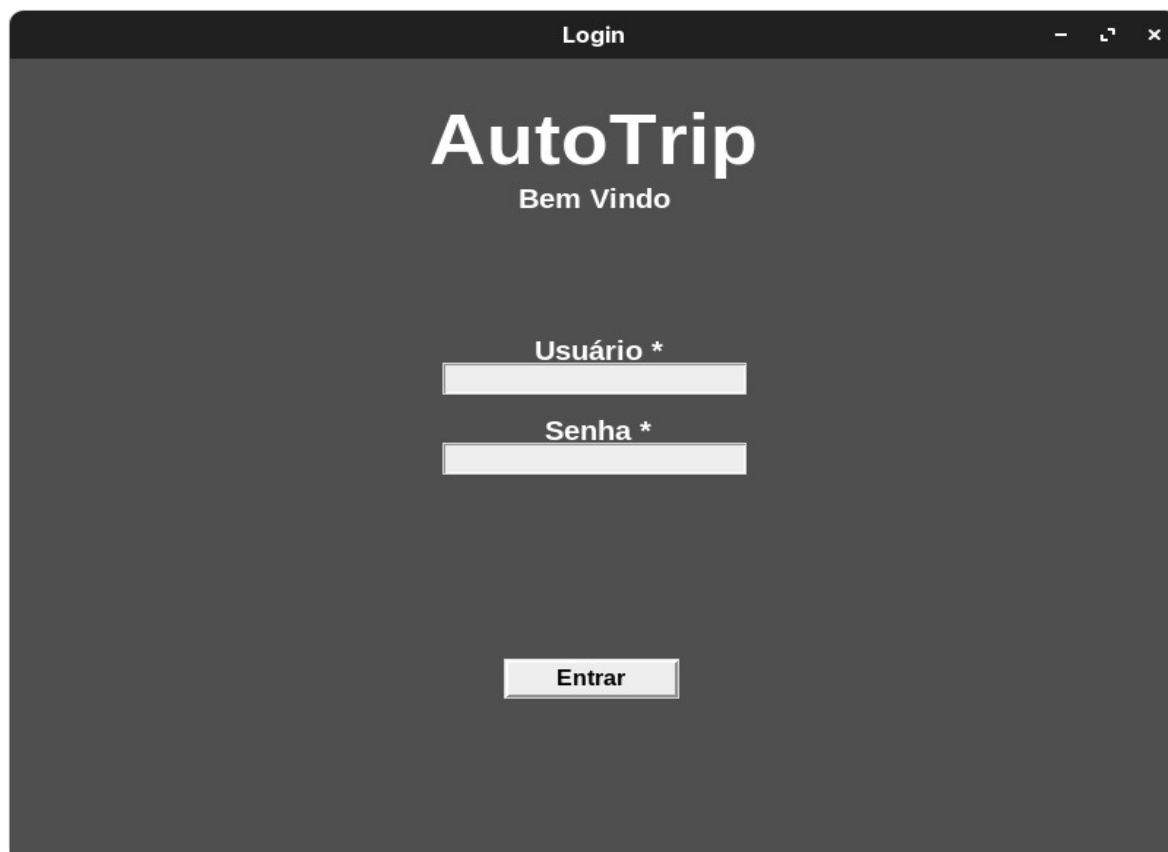
Os objetivos iniciais ainda não alcançados são: a ferramenta de alerta do sistema para manutenções agendadas, adição de uma tabela e os processos para registro dos procedimentos de manutenções mais comuns, adição uma tabela com as marcas de veículos mais populares, o relacionamento de todas as tabelas do banco de dados, implementação de *Menu Options* e máscaras de formatação de dados nos campos de entrada em cada tela. Todas essas alterações serão feitas em atualizações futuras do sistema.

Desta-se ainda o futuro promissor do projeto que não deverá se encerrar na entrega final do TG, e continuará sendo desenvolvido e aperfeiçoado para se tornar um aplicativo completo para ser distribuído por licenciamento gratuito no futuro.

As seções seguintes apresentam detalhes dos processos de negócio implementados no protótipo.

5.1 Login

A tela de Login (Fig. 5), tem como objetivo autenticar um usuário com a permissão de inserir ou alterar registros no sistema. Para facilitar e aumentar a eficiência do desenvolvimento foi implementada a janela apenas ilustrativa que ainda não valida o usuário.

Figura 5 – Tela de LoginA imagem mostra uma interface de login para um sistema chamado "AutoTrip". No topo, há uma barra de título com o texto "Login" e ícones de minimizar, maximizar e fechar. O corpo da interface tem um fundo cinza escuro. No centro, o nome "AutoTrip" é exibido em uma fonte grande e branca, com "Bem Vindo" logo abaixo em uma fonte menor. Abaixo do texto, há dois campos de entrada brancos: o primeiro é rotulado "Usuário *" e o segundo "Senha *". Ambos os campos têm uma borda cinza. Abaixo dos campos, há um botão branco com o texto "Entrar" em cinza escuro.

Fonte: o autor

5.2 Consulta de veículos

Após o Login, o usuário é redirecionado à janela mostrada na Fig. 6, que tem como objetivo mostrar todos os veículos cadastrados no sistema. Nela constam dados fundamentais dos veículos, como a identificação e a data em que foi cadastrado. A partir desta tela, tem-se acesso a diversas outras que completam as funcionalidades do software.

5.3 Cadastro de veículos

Na janela da Fig. 7, que é acessível através de Busca pela placa na janela de Consulta de Veículos, tem-se acesso a todos os campos necessários para inserção de um veículo no sistema. Nesta janela todos os campos são de preenchimento obrigatório pois tem como objetivo identificar o veículo.

Figura 6 – Consulta de Veículos cadastrados

Consultar Veículos

Sair

Digite a placa do veículo

Buscar

AutoTrip

ID	Placa	Marca	Modelo	Ano	Status	Data Cadastro
1	aaa1234	Fiat	Uno	2004	Ativo	2022-04-21 11:37:33
2	bbb2222	Fiat	Marea	2000	Inativo	2022-04-21 11:37:48
3	XXX1111	Ford	Fiesta	2010	Ativo	2022-04-28 15:33:09

Cadastrar Veículo Editar Veículo Inserir Manutenção Ver Manutenções Calculadora

Fonte: o autor

Figura 7 – Cadastrar Veículos

Buscar Veículos

Voltar

AutoTrip
Cadastrar

Placa *

Marca *

Modelo *

Ano *

Status *

Calculadora Cadastrar

Campos destacados são obrigatórios.

Fonte: o autor

5.4 Editar Veículo

Nesta janela (Fig. 8), é disponibilizada a edição de dados de um veículo já existente no sistema. Erros na hora de inserir um veículo podem acontecer, e aqui é o lugar onde o usuário pode corrigi-los. Nesta janela, por ser possível a alteração da identificação do veículo, permanece o requisito de todos os campos serem de preenchimento obrigatório.

Além de corrigir possíveis erros na inclusão de um veículo, nesta janela também pode-se editar o Status de um veículo, onde Ativo significa que o veículo está em operação na frota, e Inativo que este veículo não é mais de propriedade da empresa.

Figura 8 – Editar Veículos cadastrados

Fonte: o autor

5.5 Inserir uma manutenção

Nesta janela (Fig. 9), que é a mais importante do sistema, o usuário pode registrar ou agendar uma manutenção de um veículo no sistema. Por meio dos

campos disponibilizados, ele pode registrar desde um abastecimento, até programar uma futura manutenção.

Para fins de funcionalidade geral do sistema, nesta janela é obrigatório o preenchimento dos campos Manutenção e KM Atual, pois é através desses campos que o sistema, futuramente, alertará o usuário sobre uma manutenção a ser realizada ou que tenha sido previamente agendada no sistema.

Para o agendamento de uma manutenção, esta janela fornece duas opções. A primeira opção é pelo campo Próxima Manutenção (KM), onde o usuário insere a quilometragem na qual será feita a próxima manutenção, e através de futuros processos com base no KM Atual (por isso é um campo obrigatório), o sistema irá calcular e alertar o usuário caso o veículo tenha chegado na quilometragem correspondente a uma manutenção programada previamente.

A segunda opção de agendamento, é pelo campo Próxima Manutenção (Data), onde será inserida, pelo usuário, uma data futura para a manutenção.

Figura 9 – Atualizar dados de Manutenção Veículos

Atualizar Veículos

Voltar

AutoTrip
Atualizar

Placa	aaa1234	Manutenção	
Marca	Fiat	KM Atual	
Modelo	Uno	Próxima Manutenção (KM)	
Ano	2004	Próxima Manutenção (Data)	

Observação

Calculadora

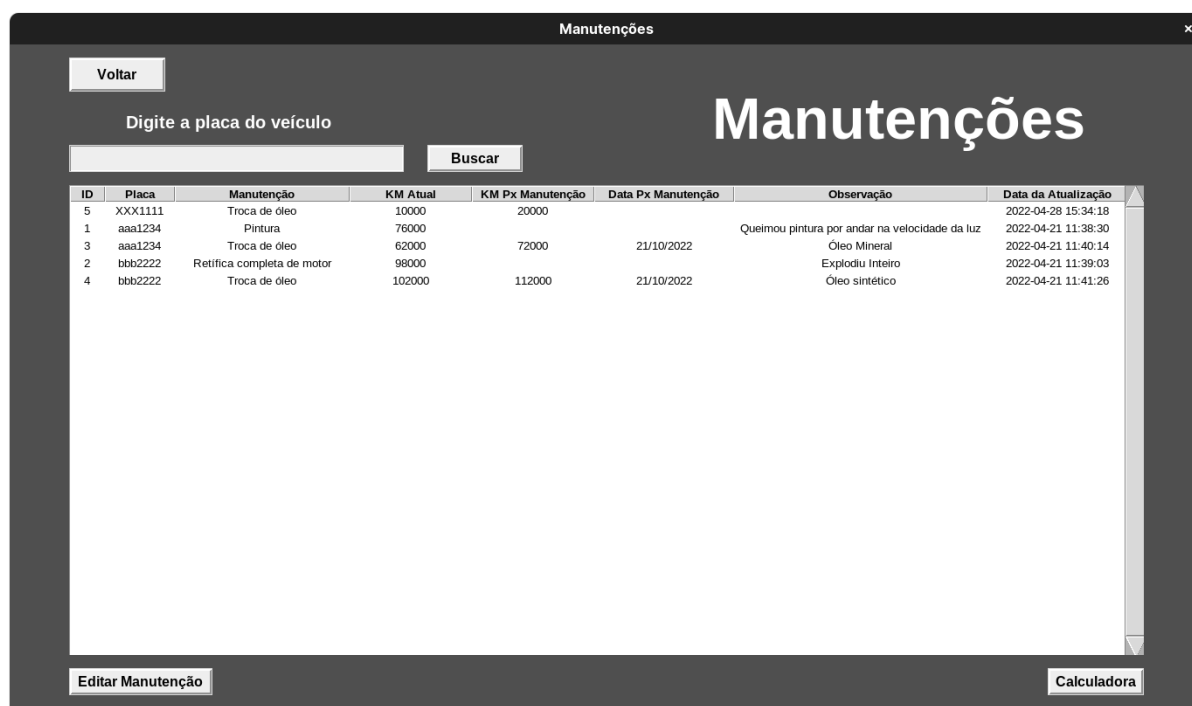
Atualizar

Fonte: o autor

5.6 Manutenções

Nesta janela (Fig. 10), o usuário tem acesso a todas as manutenções realizadas e manutenções agendadas através de uma lista. Nesta janela existe a ferramenta de Busca, onde as manutenções são filtradas de acordo com a Placa do veículo. Sendo assim, sempre que o usuário tiver interesse em consultar alguma manutenção realizada, ou alguma manutenção agendada, basta inserir a Placa do veículo desejado no campo de busca e pressionar o botão Buscar.

Figura 10 – Manutenções de Veículos



ID	Placa	Manutenção	KM Atual	KM Px Manutenção	Data Px Manutenção	Observação	Data da Atualização
5	XXX1111	Troca de óleo	10000	20000			2022-04-28 15:34:18
1	aaa1234	Pintura	76000			Queimou pintura por andar na velocidade da luz	2022-04-21 11:38:30
3	aaa1234	Troca de óleo	62000	72000	21/10/2022	Óleo Mineral	2022-04-21 11:40:14
2	bbb2222	Retífica completa de motor	98000			Explodiu Inteiro	2022-04-21 11:39:03
4	bbb2222	Troca de óleo	102000	112000	21/10/2022	Óleo sintético	2022-04-21 11:41:26

Fonte: o autor

5.7 Editar Manutenção

Assim como na janela de Consultar Veículos, nesta janela de Manutenções (Fig. 11) também existe a possibilidade de editar e corrigir algum registro que possa ter sido inserido errado pelo usuário. Esta janela segue as mesmas regras da janela de inserir manutenção, onde os campos Manutenção e KM Atual são de preenchimento obrigatório.

Figura 11 – Edição de Manutenções de Veículos

A imagem mostra uma interface web para edição de manutenção de veículos. No topo, há uma barra de título com o texto "Editar Veículo" e um ícone de fechar. Abaixo, no canto superior esquerdo, há um botão "Voltar". O título principal "AutoTrip" está centralizado, com o subtítulo "Editar Manutenção" logo abaixo. O formulário contém os seguintes campos:

- Manutenção:** Um campo de texto com o valor "5" e um campo de texto com o valor "Troca de óleo".
- KM Atual:** Um campo de texto com o valor "10000".
- Próxima Manutenção (KM):** Um campo de texto com o valor "20000".
- Próxima Manutenção (Data):** Um campo de texto vazio.
- Observação:** Um campo de texto vazio.

No canto inferior esquerdo, há um botão "Calculadora". No canto inferior direito, há um botão "Editar".

Fonte: o autor

Considerações finais

Os desafios encontrados durante o desenvolvimento deste projeto me possibilitaram tornar um profissional melhor, com uma visão muito mais realista da análise e desenvolvimento de software, área que vou atuar no futuro. Dentre estes desafios, ressalto a coragem em iniciar um projeto do zero, com um estilo de programação completamente novo e usando ferramentas que nunca havia usado antes.

Este projeto não termina aqui. As aplicações *web* são uma realidade e estão se tornando cada vez mais poderosas e acessíveis. Por este motivo, este projeto passará por uma transição de sua arquitetura *desktop* para uma arquitetura multiplataformas muito mais simples de ser utilizada. Esta migração possibilitará o projeto a alcançar voos muito mais altos, se tornando um produto muito mais completo para o mercado de softwares. Em relação a acessibilidade do usuário e alcance de

mercado, o sistema também irá passar por traduções, almejando entrar no disputado mercado internacional.

Deixo registrado neste documento minha gratidão aos professores da Fatec Franca Dr. Thomaz Novelino, que durante estes três anos foram tão importantes durante a minha formação profissional e meu crescimento pessoal.

Serei também grato por todos os desafios que passei durante este período, onde infelizmente perdemos amigos e parentes por conta de uma pandemia global, mas que, a mim, serviu para aumentar o sentimento de esperança de que evoluiremos como seres humanos melhores a cada dia. Encerro este ciclo na minha vida com a absoluta certeza de que crescemos perante as dificuldades.

Referências

TROVÃO, RENYERE. **Vida nova ao carro: retífica é a opção para deixar o motor 'zero de fábrica'**. 2018. Disponível em:

<<https://www.gazetadopovo.com.br/automoveis/vida-nova-ao-carro-retifica-e-a-opcao-para-deixar-o-motor-zero-de-fabrica-7qz7tq7i7p39z9ol8bndu2mg6/#:~:text=O%20valor%20de%20ret%C3%Adfica%20depende,o%20custo%20sobe%20mais%20ainda>>. Acesso em: 16.fev.2022.

VOLVO. **Descrição do cabeçote para caminhões Volvo Reman**. sd. Disponível em:

<https://www.volvopecas.com.br/cabecote_para_caminhoes_volvo_85022751-/p>. Acesso em: 16.fev.2022.