UNIVERSIDADE DE RIBEIRÃO PRETO

ENGENHARIA DA COMPUTAÇÃO

FERNANDO MOGNO VOLPINI - 840608

GABRIEL SANTOS DE ALMEIDA - 839842

CHECK-UP CAR

RIBEIRÃO PRETO

2024

FERNANDO MOGNO VOLPINI - 840608

GABRIEL SANTOS DE ALMEIDA - 839842

CHECK-UP CAR

Relatório sobre o projeto check-up car do desafio das Engenharias do curso de Engenharia da Computação.

Docentes: Prof. Me. Otavio Marson Junior e Prof. Me [Rodrigo Plotze.](https://www.linkedin.com/in/rodrigo-plotze-6502b434/overlay/about-this-profile/)

.

Ribeirão Preto

2024

SUMÁRIO

[1 OBJETIVO ........................................................................................................................... 4](#Objetivo)

[2 FUNDAMENTOS TEÓRICOS .......................................................................................... 4](#Fundamentos)

[3 MATERIAS UTILIZADOS ................................................................................................ 9](#Materiais)

[4 DADOS EXPERIMENTAIS ............................................................................................... 9](#Dados)

[5 PROCEDIMENTO ............................................................................................................ 10](#Procedimento)

[6 RESULTADOS .................................................................................................................. 10](#Resultados)

[7 CONCLUSÃO .................................................................................................................... 10](#Conclusão)

[REFERÊNCIAS .................................................................................................................... 10](#Referências)

1 OBJETIVO

Desenvolver uma base de dados para armazenar informações de motoristas e veículos, enviando alertas automáticos para a realização de revisões, garantindo segurança e desempenho dos veículos.

2 FUNDAMENTOS TEÓRICOS

Introdução

O projeto Check-Up Car visa criar um sistema de base de dados que armazene informações sobre motoristas e seus veículos, com o objetivo de enviar alertas sobre revisões veiculares. Este sistema será uma ferramenta importante para ajudar motoristas a manterem seus veículos em boas condições, garantindo segurança e desempenho adequados.

1. Importância da Manutenção Preventiva de Veículos

A manutenção preventiva de veículos é crucial para garantir a segurança nas estradas e a longevidade dos automóveis. Manutenções regulares ajudam a identificar e corrigir problemas antes que se tornem sérios, evitando falhas mecânicas e acidentes. Além disso, manter o veículo em boas condições pode melhorar a eficiência de combustível e reduzir os custos de reparos a longo prazo.

2. Estrutura da Base de Dados

Para implementar o sistema Check-Up Car, é necessário desenvolver uma base de dados robusta que possa armazenar e gerenciar várias informações críticas sobre os veículos e seus motoristas.

3. Algoritmos de Alerta

Os algoritmos de alerta são fundamentais para o funcionamento do Check-Up Car. Eles utilizam dados históricos e predefinidos para determinar quando um veículo precisa de manutenção. Alguns critérios comuns incluem:

Quilometragem Percorrida: Alertas podem ser configurados para disparar quando o veículo atingir uma quilometragem específica desde a última revisão.

Intervalo de Tempo: Se um certo período (por exemplo, seis meses) passou desde a última manutenção, o sistema pode enviar um lembrete para uma nova revisão.

Histórico de Manutenções: Analisando o histórico de manutenções, o sistema pode prever possíveis problemas recorrentes e sugerir verificações adicionais.

4. Tecnologias Utilizadas

Biblioteca

Esse código utiliza a biblioteca flet para a criação de interfaces gráficas em Python. flet parece ser uma biblioteca fictícia, pois não é uma biblioteca conhecida no ecossistema Python até o meu último treinamento em janeiro de 2022. O código também importa o módulo datetime da biblioteca padrão do Python, mas não o utiliza no trecho mostrado.

Framework

Flutter é um framework de código aberto do Google para criar aplicativos nativos para dispositivos móveis, web e desktop com uma única base de código, usando a linguagem Dart. Ele oferece um conjunto abrangente de widgets e ferramentas para desenvolver interfaces de usuário bonitas e responsivas, com destaque para sua rapidez de desenvolvimento, desempenho sólido e capacidade de hot reload.

Em Python, a declaração import é usada para importar módulos, pacotes ou itens específicos de um módulo. No caso do Flutter, um framework para desenvolvimento de aplicativos móveis, não é importado diretamente em Python como um pacote convencional. Em vez disso, você instala o SDK do Flutter e utiliza a CLI para desenvolver seus aplicativos.

Linguagem de Programação

Python

5. Benefícios do Sistema

Segurança

Garantir que os veículos estejam sempre em boas condições de funcionamento ajuda a prevenir acidentes causados por falhas mecânicas.

Economia

Manutenções preventivas regulares podem evitar reparos caros decorrentes de negligência e aumentar a vida útil do veículo.

Conveniência

Os alertas automatizados facilitam a vida dos motoristas, lembrando-os das manutenções necessárias e ajudando-os a agendar revisões de forma conveniente.

3 MATERIAS UTILIZADOS

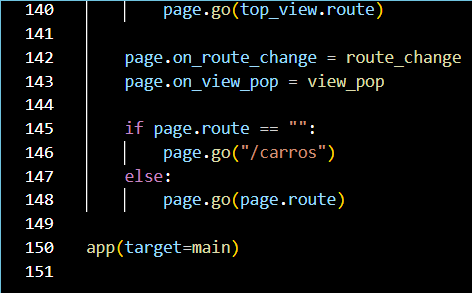
As tecnologias utilizadas no projeto incluem o uso da biblioteca fictícia "flet" para criar interfaces gráficas em Python. O código também importa o módulo "datetime" da biblioteca padrão do Python, embora não o utilize. Além disso, menciona-se o framework Flutter, do Google, para desenvolver aplicativos nativos com uma única base de código em Dart.

4 DADOS EXPERIMENTAIS

Codigo Python – Terminal



Codigo Python – Interface grafica



5 PROCEDIMENTO

Para realizar esse código em Python, foi utilizado a biblioteca flet, que é uma biblioteca para criação de interfaces gráficas de usuário (GUI) em Python. Aqui está uma explicação detalhada do procedimento para a realização desse código:

Instalação da biblioteca: Antes de começar, é necessário instalar a biblioteca flet usando o pip, o gerenciador de pacotes do Python. Isso pode ser feito com o seguinte comando no terminal:

pip install flet

Importação de bibliotecas e módulos necessários: Para utilizar a biblioteca flet, é preciso importá-la no início do código, juntamente com os ícones e os tipos de alinhamento necessários para a formatação da interface:

import flet as ft from flet import \* import datetime

Definição da função de troca de abas: Foi definida a função trocando(e) que será chamada quando houver uma mudança de aba no componente Tabs. Neste caso, a função não faz nada (pass), mas normalmente seria utilizada para realizar alguma ação específica ao trocar de aba.

Criação do componente Tabs: O componente Tabs foi criado com três abas: "Carros", "Adicionar Veículo" e "Verificar Revisão". Cada aba tem um texto e um ícone associado.

Definição da função principal (main): A função main(page: ft.Page) é o ponto de entrada para a criação da interface gráfica. Ela recebe um objeto Page como argumento, que representa a janela da aplicação.

Criação da barra superior (minha\_barra): Foi criada uma barra superior com um gradiente de cores, ícones de menu, título e abas. A barra foi configurada com um layout de colunas e linhas, utilizando os componentes Row e Column da biblioteca flet.

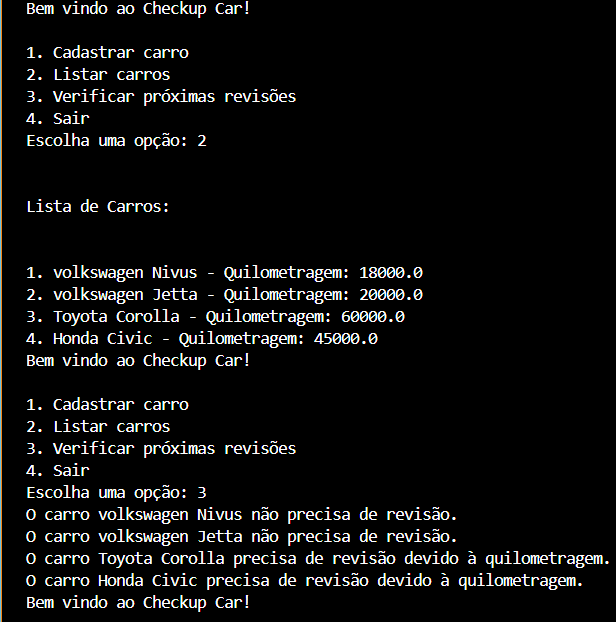
Adição da barra à janela: A barra superior foi adicionada à janela da aplicação usando o método overlay.append(minha\_barra).

Definição do conteúdo da página: Foi definido o conteúdo principal da página, que neste caso está vazio. Normalmente, aqui seria adicionado o conteúdo específico de cada aba, como formulários, tabelas ou outros componentes.

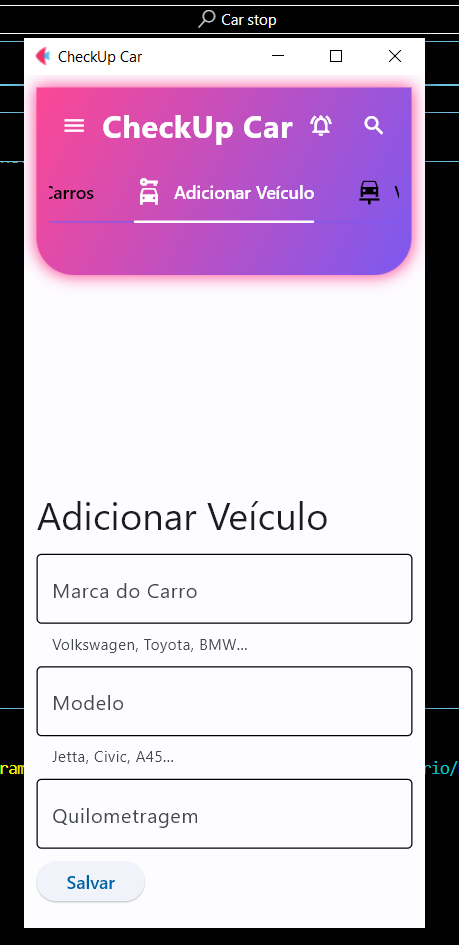
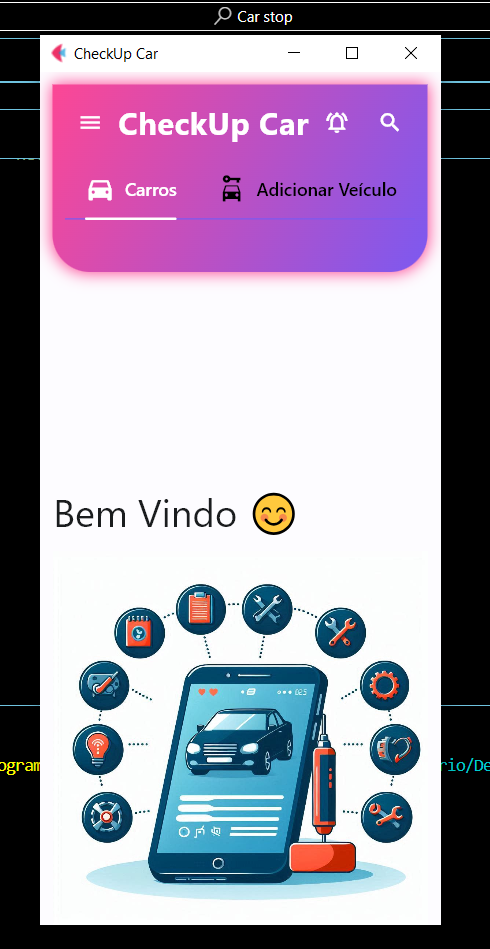
Execução da aplicação: Por fim, a aplicação foi executada usando o método ft.app(target=main), passando a função main como argumento para iniciar a interface gráfica.

6 RESULTADOS

Codigo Python – Terminal



Codigo Python – Interface gráfica



7 CONCLUSÃO

O projeto Check-Up Car propõe um sistema que não só melhora a manutenção de veículos, mas também aumenta a segurança nas estradas e a satisfação dos motoristas. Com a implementação de uma base de dados eficiente e algoritmos de alerta precisos, este sistema pode se tornar uma ferramenta indispensável para todos os motoristas preocupados com a manutenção de seus veículos.

REFERÊNCIAS

<https://phylos.net/2023-07-10/python-com-flet>

<https://pypi.org/project/flet/>

<https://python.org.br/comunidades-locais/>

<https://docs.python.org/pt-br/3/tutorial/classes.html>

<https://www.pythonprogressivo.net/2018/11/Metodos-Metodo-init-Atributos.html>

<https://kinsta.com/pt/blog/programacao-orientada-objetos-python/>

<https://github.com/josemalcher/Aprenda-Programacao-Orientada-a-Objeto-em-21-dias>

<https://github.com/Alliedium/awesome-software-engineering>

<https://cursos.alura.com.br/loginForm>

<https://www.udemy.com/?utm_source=aff-campaign&utm_medium=udemyads&LSNPUBID=0EOJOrTo2D4&ranMID=47901&ranEAID=0EOJOrTo2D4&ranSiteID=0EOJOrTo2D4-qaecJCJCFmoO_ZcotEYvRA>

https://www.flet.dev