

# ChevoTech

Integrantes:

RM 99404 - Thiago Garcia Tonato

RM 555502 - Ian Madeira Gonçalves da Silva

RM 555109 - Murilo Ribeiro Santos

# Sumário

1. Introdução
2. Código Fonte
3. Funcionalidades
4. Introduções de uso
5. Requisitos do sistema

# Introdução do projeto

A solução elaborada foi um conjunto de sensores integrados a um módulo de coleta de dados automotivos. O objetivo deste projeto é oferecer uma solução que permita monitorar diversos parâmetros do veículo, como temperatura, sonda lambda, desgaste dos freios, entre outros, de forma automatizada, eficiente e sem a necessidade de ação humana.

# Código Fonte

O código fornecido é escrito em Python e implementa um sistema de coleta e gerenciamento de dados automotivos com base em um conjunto de sensores, permitindo aos usuários interagir com as informações dos sensores e realizar operações relacionadas à manutenção do veículo.

# Funcionalidades

**Cadastro de usuário:** O código permite que novos usuários se cadastrem inserindo um nome de usuário e uma senha. Essas informações são armazenadas nas variáveis `nome` e `senha`. Após o cadastro bem-sucedido, a variável booleana `cadastro` é definida como verdadeira.

**Login de Usuário:** Após o cadastro, os usuários podem fazer login fornecendo seu nome de usuário e senha. O sistema verifica se as credenciais estão corretas e permite o acesso ao menu principal. Se as credenciais estiverem incorretas ou se o usuário não estiver cadastrado, uma mensagem de login inválido é exibida.

**Coleta de Dados dos Sensores:** A funcionalidade de coleta de dados dos sensores permite ao usuário simular a obtenção de dados dos sensores. Utilizando a biblioteca `numpy`, são gerados números aleatórios que simulam dados de sensores automotivos. Esses dados são então armazenados em uma lista chamada `dados_sensores`.

**Previsão de Manutenção:** Com base nos dados coletados dos sensores, o sistema realiza uma previsão sobre a necessidade de manutenção. Calcula-se a média dos dados dos sensores e compara-se com um `threshold` predefinido. Se a média dos dados exceder esse `threshold`, é indicada a necessidade de manutenção. Caso contrário, é exibida uma mensagem informando que nenhuma manutenção é necessária.

**Adição da Data de Última Modificação:** Esta funcionalidade permite ao usuário adicionar a data da última manutenção realizada. O usuário é solicitado a inserir a data no formato "dd/MM/yyyy", que é então adicionada a uma lista chamada `lista_manutencoes`.

**Visualização do Histórico de Manutenções:** O sistema permite ao usuário visualizar todas as datas das manutenções previamente adicionadas. Essas datas são exibidas a partir da lista `lista_manutencoes`.

# Introduções de Uso

## 1. Cadastro de Usuário:

- Ao iniciar o programa, escolha a opção "Fazer Cadastro" no menu principal.
- Insira um nome de usuário quando solicitado.
- Insira uma senha quando solicitado.
- Após o cadastro bem-sucedido, você será notificado com a mensagem "Cadastro realizado!".

## 2. Login de Usuário:

- Escolha a opção "Fazer login" no menu principal.
- Insira o nome de usuário e a senha que você cadastrou anteriormente.
- Se as credenciais estiverem corretas, você será direcionado ao menu principal. Caso contrário, uma mensagem de "Login inválido!" será exibida.

## 3. Coleta de Dados dos Sensores:

- Após fazer login, escolha a opção "Coletar dados dos sensores" no menu principal.
- Dados dos sensores serão simulados e coletados automaticamente.
- Você receberá uma mensagem indicando que os dados dos sensores foram coletados com sucesso.

## 4. Previsão de Manutenção:

- Selecione a opção "Prever manutenção" no menu principal.
- O sistema calculará a média dos dados dos sensores coletados.
- Com base nessa média, uma mensagem será exibida indicando se é recomendada uma manutenção ou se nenhuma manutenção é necessária.

## 5. Adição da Data da Última Manutenção:

- Escolha a opção "Adicionar data da manutenção" no menu principal.
- Insira a data da última manutenção no formato "dd/MM/yyyy" quando solicitado.
- A data será adicionada à lista de manutenções realizadas.

## 6. Visualização do Histórico de Manutenções:

- Selecione a opção "Histórico de manutenção" no menu principal.
- Todas as datas de manutenção previamente adicionadas serão exibidas.

## 7. Encerramento do Programa:

- A qualquer momento, escolha a opção "Sair" no menu principal para encerrar o programa.
- Uma mensagem de "LogOut feito! Obrigado!" será exibida antes de encerrar o programa.

# Requisitos do sistema

## 1. Python:

- O sistema requer a instalação do interpretador Python na máquina do usuário.
- Versão recomendada: Python 3.x.

## 2. Biblioteca Numpy:

- O código faz uso da biblioteca NumPy para geração de números aleatórios simulando dados dos sensores.
- Certifique-se de que a biblioteca NumPy esteja instalada na máquina do usuário.
- Caso não esteja instalada, ela pode ser instalada via pip:

```
pip install numpy
```

# Processo de Operação Simples

1

## Iniciar Programa

O usuário abre o programa e é recebido com um menu de cadastro e login.

2

## Cadastro

O usuário se cadastra com login e senha.

3

## Login

O usuário inicia login com os dados feitos no cadastro.

4

## Menu Principal

Após o login aparece o menu principal com as opções dadas.



# Código-fonte

Código-fonte comentado para melhor entendimento do projeto.

```
1 import numpy as np
2
3 #Setando variáveis necessárias para a entrada ao programa
4 nome = ''
5 senha = ''
6 cadastro = False
7
8 #Menu com repetição para login
9 while True:
10     print('')
11     print('>>>CHEVOTECH<<<')
12     print('-----')
13     print('1. Fazer Cadastro')
14     print('2. Fazer login')
15     print('3. Sair')
16     print('-----')
17
18     opcao_login = int(input("Digite o número da opção que deseja: "))
19
20     #Interações com o menu
21     match opcao_login:
22
23         #Cadastrar
24         case 1:
25             print('')
26             print('>>>CHEVOTECH<<<')
27             print('-----')
28             nome_cadastro = input('Digite seu login para cadastro: ')
29             nome = nome_cadastro
30             senha_cadastro = input('Digite sua senha para cadastro: ')
31             senha = senha_cadastro
32             cadastro = True
33             print('Cadastro realizado!')
34
35         #Login
36         case 2:
37             if cadastro:
38                 print('')
39                 print('>>>CHEVOTECH<<<')
40                 print('-----')
41                 nome_login = input('Digite seu login: ')
42                 senha_login = input('Digite sua senha: ')
43
44                 if nome_login == nome and senha_login == senha:
45                     lista_manutencoes = []
46                     dados_sensores = []
47
48                     #Menu principal do programa
49                     while True:
50                         print('')
51                         print('>>>CHEVOTECH<<<')
52                         print('-----')
53                         print('Bem vindo! O que deseja fazer?')
54                         print('1. Coletar dados dos sensores')
55                         print('2. Prever manutenção')
56                         print('3. Adicionar data da manutenção')
57                         print('4. Histórico de manutenção')
58                         print('5. Sair')
59
60                         opcao_menu = input('Digite o número da opção que deseja: ')
61
62                         match opcao_menu:
63                             #Coletar dados do sensor (usado np.random para gerar numeros fictícios)
64                             case '1':
65                                 print('')
66                                 dados_sensores.extend(np.random.rand(10))
67                                 print('Dados dos sensores coletados com sucesso!')
68
69                             #Prever manutenção necessária com base nos dados gerados pelo sensor
70                             case '2':
71                                 threshold = 0.8
72                                 print('')
73                                 if np.mean(dados_sensores) > threshold:
74                                     print('Manutenção recomendada.')
75                                 else:
76                                     print('Nenhuma manutenção necessária!')
77
78                             #Adicionar ultima manutenção feita a uma lista
79                             case '3':
80                                 print('')
81                                 manutencao = input('Adicione a data da manutenção feita (dd/MM/yyyy): ')
82                                 lista_manutencoes.append(manutencao)
83
84                             #Mostrar lista de todas as manutenções feitas
85                             case '4':
86                                 print('')
87                                 print(f'Manutenções feitas nos dias: {lista_manutencoes}')
88
89                             #Sair do programa
90                             case '5':
91                                 print('')
92                                 print('LogOut feito! Obrigado!')
93                                 break
94
95                             #Caso nenhuma opção for válida
96                             case _:
97                                 print('')
98                                 print('Digite uma opção válida!')
99
100
101                     else:
102                         print('Login inválido!')
103
104             else:
105                 print('')
106                 print('Realize o cadastro para fazer login!')
107
108         #Sair do programa
109         case 3:
110             print('Obrigado!')
111             break
112
113         #Caso nenhuma opção for válida
114         case _:
115             print('')
116             print('Digite uma opção válida.')
```