

**Faculdade de Informática e Administração
Paulista**

**Anderson de Sousa Pedro, Luigi Giuseppe Jampietro Sciarretta e Vinicius
Saes de Souza**

**Computational Thinking Using
Python - AutoBot**

São Paulo

2024

Sumário

1 - Objetivos Autobot	3
2 - Código Fonte	4

1 - Objetivos Autobot

O Autobot busca uma solução inovadora para revolucionar e simplificar a manutenção automotiva, utilizando inteligência artificial. Nosso sistema visa fornecer ao usuário uma experiência completa e integrada, desde o diagnóstico dos problemas de seu automóvel até a solução e, posteriormente, o orçamento confiável que ele poderá levar até a oficina de sua preferência, todos esses serviços dentro de um único ambiente digital.

Dentro do Autobot, temos como objetivo principal oferecer o diagnóstico assistido por chatbot. Por meio de uma simples conversa, o usuário pode descrever os sintomas e assim diagnosticar problemas em seus veículos de forma rápida e precisa. Após o diagnóstico, o Autobot será capaz de fornecer um orçamento estimado para o reparo do veículo, baseado no preço das peças necessárias, que estarão cadastradas e disponíveis para consulta dentro do nosso sistema. Isso proporcionará aos usuários transparência e clareza sobre os custos envolvidos.

Além desses serviços, o nosso sistema também terá como objetivo facilitar o gerenciamento de veículos. Os usuários poderão criar um perfil personalizado, onde podem cadastrar informações sobre seus veículos, tornando o processo de monitoramento e manutenção mais eficiente. Ligado ao cadastro de veículos, teremos dois outros serviços.

O primeiro será o diário de bordo personalizado, uma funcionalidade exclusiva do Autobot, que fornece aos usuários lembretes e sugestões sobre as principais verificações e manutenções preventivas a serem realizadas em seus veículos, garantindo sua segurança e prolongando sua vida útil.

A segunda função ligada ao cadastro de veículo são os manuais de manutenção e reparo, esta área focada para os entusiastas que desejam realizar reparos por conta própria. O Autobot oferece uma seção dedicada a manuais de reparo. Aqui, os usuários encontrarão orientações passo a passo e dicas úteis para realizar reparos básicos em seus veículos, tudo de forma segura e eficiente.

Por último, nosso último objetivo é oferecer uma função de consulta de preços de peças, permitindo aos usuários comparar e encontrar os melhores preços para os componentes necessários para a manutenção de seus veículos.

O Projeto Autobot visa tornar a mecânica automotiva conveniente e transparente para todos os proprietários de veículos. Ao combinar tecnologia de ponta com uma interface amigável, o Autobot está preparado para revolucionar a forma como lidamos com os cuidados de nossos carros.

2 – Mudanças no código:

O segundo código representa uma significativa evolução em relação ao primeiro. Introduzindo novas funcionalidades, como limpeza de tela, exibição de opções e consulta de veículos, o programa se tornou mais robusto e amigável. Uma abordagem mais estruturada, dividindo o código em partes menores, contribui para uma leitura mais fluída e compreensão facilitada.

Além disso, foram adicionadas informações detalhadas sobre problemas automotivos, acompanhadas por um menu mais sofisticado, elevando a apresentação a um nível mais elegante. A validação dos dados de entrada foi aprimorada com a inclusão de funções especializadas para verificar se são palavras ou números, garantindo a integridade dos dados fornecidos pelo usuário.

A inclusão da funcionalidade de consulta de veículos adiciona mais utilidade ao programa, permitindo ao usuário revisar os veículos cadastrados de forma simples e direta. Caso nenhum veículo tenha sido registrado ainda, uma mensagem informativa é exibida, mantendo o usuário informado.

Após o cadastro de um novo veículo, o usuário tem a opção de verificar os veículos já registrados, cadastrar outro veículo ou retornar ao menu principal, aumentando a flexibilidade e praticidade do programa.

Simultaneamente, foram removidos elementos desnecessários do código, resultando em uma estrutura mais enxuta e fácil de compreender. A substituição do método anterior de limpar a tela usando `os.system('cls')` por uma nova função mais eficiente é um exemplo disso. Ademais, o código foi refinado para eliminar redundâncias e simplificar a lógica, tornando-o mais conciso e legível.

3 – Código Fonte:

```
import os
lista_numeros = '0123456789'

veiculos = []
problemas_painel = ['Superaquecimento', 'Pane elétrica', 'Problemas no câmbio', 'Bateria ruim', 'Falta de combustível']
itens_diario_bordo = ['alinhamento dos pneus', 'calibração dos pneus', 'nível do óleo', 'luzes do carro']
atributos_carro = ['Fabricante', 'Modelo', 'Ano', 'Placa']

problemas_info_superaquecimento = """
O problema ocorre quando não há uma circulação de água adequada ou qualquer outro defeito que interfira no funcionamento do sistema de arrefecimento. Apesar de exigir o acompanhamento de um profissional, há algumas ações que você pode tomar para se prevenir. Confira:

- verifique o ponteiro que marca a temperatura. Se ele chegar no vermelho, leve o veículo direto para uma oficina;
- complete o nível de água do motor. Porém, caso tenha que fazer isso diversas vezes em um curto período de tempo, pode ser um aviso de que há algo errado;
- fique atento às mangueiras internas. Caso alguma esteja estufada, pode haver uma má circulação da água."""

problemas_info_pane_eletrica = """
A pane elétrica exige um cuidado muito especial, afinal de contas, por mais conhecimento que o motorista tenha, ele dificilmente será capaz de prever problemas com o alternador ou com a bomba de combustível, por exemplo. Contudo, há medidas que o dono do veículo pode tomar para se prevenir, tais como:

- verificar se há sinais de dificuldade para ligar o automóvel;
- certificar-se de que não está acontecendo nenhum vazamento de ácido;
- observar se acontece alguma redução das luzes ao dar a partida no carro;
- conferir se a correia do alternador não está esbranquiçada ou desfiando.

Além disso, é importante mencionar que, se você for “tunar” o carro inserindo DVDs, caixas potentes de som e demais acessórios, é preciso
```

preparar a bateria e o alternador para isso. Caso isso não seja feito, o veículo pode parar de funcionar e te deixar na mão.

"""

`problemas_info_problemas_cambio = ""`

Defeitos no conjunto do câmbio ao longo do tempo são comuns e inevitáveis. A melhor saída é ter atenção aos sinais de que algo não está certo com a transmissão para que seja possível corrigir antecipadamente. Veja o que observar para descobrir se há problemas no câmbio:

- A marcha arranha antes de engatar?
- Existe um grilo que some ao pisar na embreagem?
- As marchas escapam?
- Há trepidação ao arrancar?
- A alavanca se movimenta?

Quanto aos cuidados para evitar problemas como esses, podemos afirmar que fazer a revisão é primordial, uma vez que o nível de óleo não pode estar mais baixo que o indicado, além de precisar ser trocado, pois tem data de validade. A troca de óleo vai garantir o perfeito funcionamento e a vida útil da caixa de marcha. Também há alguns fatores que colaboram para um desgaste excessivo, e você deve evitá-los. Confira:

- não apoie a mão no câmbio. Use a alavanca apenas na troca de marchas;
- evite deixar o pé sobre a embreagem enquanto estiver dirigindo;
- evite arrancar em segunda marcha. O motor suporta, mas isso causa um desgaste demasiado.

"""

`problemas_info_bateria = ""`

A maioria das baterias de automóveis deve durar aproximadamente 3 anos ou então 80 mil km. Uma bateria descarregada é, em geral, causada por amperes que se reduzem naturalmente à medida que o produto perde sua capacidade de manter uma carga.

Um sensor de temperatura da bateria, o alternador danificado ou outros componentes do sistema de carregamento podem acelerar esse problema. O ideal, portanto, é substituir a bateria do seu veículo no período indicado pelo fabricante, mesmo que não esteja apresentando sinais de danos.

"""

`problemas_info_falta_combustivel = ""`

A falta de combustível pode não acarretar grandes prejuízos ao automóvel, porém, é um problema difícil de ser solucionado no meio da estrada, devido à pequena quantidade de postos de combustíveis nas rodovias. Para que você não corra o risco de ficar no meio do caminho, é fundamental seguir à risca estas dicas:

certifique-se de que o marcador do automóvel está funcionando bem. Caso o ponto demore para descer, ou se está descendo rápido demais, talvez haja um problema na bomba;

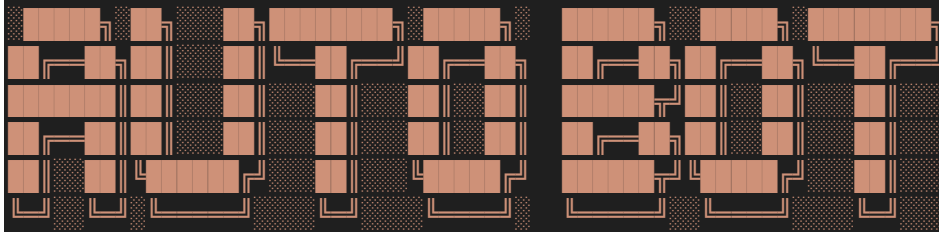
a marcação de $\frac{1}{4}$ de combustível deve ser sempre considerada como indicativo para abastecer no próximo posto;

evite postos com procedência duvidosa, afinal de contas, o produto fornecido pode estar alterado.

"""

```
def limpa_tela():  
    os.system('cls' if os.name == 'nt' else 'clear')
```

```
def tela_inicial(): #Utilizado site fsymbols.com/letters/  
    print(  
    """
```



"""

)

```
def menu_opcoes():  
    print('1. Cadastrar Veículo')  
    print('2. Painel de Problemas')  
    print('3. Diário de Bordo')  
    print('4. Consultar veículos')  
    print('5. Sair do programa')  
    nmrOpcoesMenu = 5  
    escolhe_opcao(nmrOpcoesMenu)
```

```
def exhibe_subtitulo(titulo: str):  
    limpa_tela()  
    titulo = titulo.upper()  
    linha = '*' * (len(titulo) + 6)  
    print(linha)  
    # print(titulo.ljust(22)) # ----- PQ NÃO DA O  
JUSTIFY?????????????????  
    print(titulo.center(len(linha)))  
    print(linha)
```

```
def finaliza_programa():  
    exhibe_subtitulo('..Finalizando programa..')  
    exit()
```

```

def eh_alphanumeric(s: str) -> bool:
    return s.isalnum()

def eh_int(s: str) -> bool:
    return s.isdigit()

def consultar_veiculos():
    limpa_tela()
    def escolhe_painel_consulta():
        escolha = input('\n1.Cadastrar veículo\n2.Menu
Principal\nEscolha: ')
        while not eh_int(escolha) or int(escolha) > 2:
            print('Opção inválida! digite novamente')
            escolha = input('1.Cadastrar veículo\n2.Menu
Principal\nEscolha: ')
        else:
            escolha = int(escolha)
            match escolha:
                case 1:
                    cadastrar_veiculo()
                case 2:
                    main()

    if veiculos == []:
        print("Você ainda não tem nenhum veículo cadastrado!")
    else:
        for i,item in enumerate(veiculos):
            print(f'Carro {i+1} -: {item}')
        print('\n')

    escolhe_painel_consulta()

    # match int(input('1. Voltar ao menu principal\n2. Finalizar
programa\nMenu: ')):
    #     case 1:
    #         main()
    #     case 2:
    #         finaliza_programa()

def cadastrar_veiculo():
    limpa_tela()
    exibe_subtitulo('cadastrar veiculo')
    lista_veiculo = []
    for atributo in atributos_carro:
        s = input(f'{atributo}: ')
        while not eh_alphanumeric(s) or not s.strip():
            print(f"Erro! {atributo} do veículo não pode estar vazio(a) e
não pode conter caracteres especiais (@*$-/~...).")

```



```

        s = input(f'{atributo}: ')
        lista_veiculo.append(f'{s.upper()}')
        veiculos.append(lista_veiculo)

    print("\nVeículo cadastrado com sucesso!")
    escolha = input('1. Verificar veículos cadastrados\n2. Cadastrar novo
veículo\n3. Menu principal\nEscolha: ')
    while not eh_int(escolha) or int(escolha) > 3:
        print('Opção inválida! digite novamente')
        escolha = input('1. Verificar veículos cadastrados\n2. Cadastrar
novo veículo\n3. Menu principal\nEscolha: ')
    else:
        escolha = int(escolha)
        match escolha:
            case 1:
                consultar_veiculos()
            case 2:
                cadastrar_veiculo()
            case 3:
                main()

def painel_problemas():
    limpa_tela()
    exhibe_subtitulo('painel de problemas')
    def escolhe_painel():
        escolha = input('\n1.Painel de problemas\n2.Menu
Principal\nEscolha: ')
        while not eh_int(escolha) or int(escolha) > 2:
            print('Opção inválida! digite novamente')
            escolha = input('1.Painel de problemas\n2.Menu
Principal\nEscolha: ')
        else:
            escolha = int(escolha)
            match escolha:
                case 1:
                    painel_problemas()
                case 2:
                    main()

    for i,s in enumerate(problemas_painel):
        i+=1
        print(f'{i}. {s}')
    print(f'{i+1}. Voltar ao menu principal: ')
    escolha = input('Escolha: ')
    match escolha:
        case '1':
            exhibe_subtitulo('Superaquecimento')
            print(problemas_info_superaquecimento)

```

```

        escolhe_painel()
    case '2':
        exibe_subtitulo('Pane elétrica')
        print(problemas_info_pane_eletrica)
        escolhe_painel()
    case '3':
        exibe_subtitulo('Problemas no câmbio')
        print(problemas_info_problemas_cambio)
        escolhe_painel()
    case '4':
        exibe_subtitulo('Bateria ruim')
        print(problemas_info_bateria)
        escolhe_painel()
    case '5':
        exibe_subtitulo('Falta de combustível')
        print(problemas_info_falta_combustivel)
        escolhe_painel()
    case '6':
        main()
    case _:
        painel_problemas()

def diario_bordo():
    limpa_tela()
    exibe_subtitulo('diário de bordo')
    checklist = []
    def escolhe_diario() -> str:
        escolha = input('1.Sim\n2.Não\nEscolha: ')
        while not eh_int(escolha) or int(escolha) > 2:
            print('Opção inválida! digite novamente')
            print(f'. Checou {item} ?')
            escolha = input('1.Sim\n2.Não\nEscolha: ')
        else:
            escolha = int(escolha)
            match escolha:
                case 1:
                    return '✓'
                case 2:
                    return 'X'
                case 3:
                    main()
    def escolhe_diario_menu():
        escolha = input('\n1.Refazer Checklist\n2.Menu Principal\nEscolha: ')
        while not eh_int(escolha) or int(escolha) > 2:
            print('Opção inválida! digite novamente')
            escolha = input('1.Refazer Checklist\n2.Menu Principal\nEscolha: ')

```

```

        else:
            escolha = int(escolha)
            match escolha:
                case 1:
                    diario_bordo()
                case 2:
                    main()

for i,item in enumerate(itens_diario_bordo):
    i+=1
    print(f'. Checou {item} ?')
    s = escolhe_diario()
    checklist.append(f'{item} - {s}')
print("\nCHECKLIST: ")
for i,item in enumerate(checklist):
    print(f'{i+1}. {item.capitalize()}')
escolhe_diario_menu()

def escolhe_opcao(nrItens):
    opcao = input('Menu: ')
    while True:
        if not opcao.isdigit() or int(opcao) > nrItens or int(opcao) <=
0:
            print('Erro! Digite uma opção válida do menu.')
            opcao = input('Menu: ')
        else:
            opcao = int(opcao)
            break

    match opcao:
        case 1:
            cadastrar_veiculo()
        case 2:
            painel_problemas() # A FAZER
        case 3:
            diario_bordo() # A FAZER
        case 4:
            consultar_veiculos()
        case 5:
            finaliza_programa()
        case _:
            print("Erro! digite uma opção válida")

def main():
    limpa_tela()
    tela_inicial()

```

```
menu_opcoes()
escolhe_opcao(5)

if __name__ == '__main__':
    main()
```