**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](#_Toc164769259)

[**2 - Código Fonte** 4](#_Toc164769260)

# **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**

import os

os.system('cls')

# Tela inicial com opções

print(

"""

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

  Bem-vindo(a) ao AutoBot!

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1 - Cadastrar Veículo

2 - Painel de problemas

3 - Diário de Bordo

4 - Sair do programa

"""

)

# Try catch para a escolha da opção do menu inicial

try:

    opcao\_escolhida = int(input("Digite uma opção: ")) #input da opção do menu escolhida (deve ser int)

    match opcao\_escolhida:

        case 1: #Cadastrar o veículo, colhendo input de Marca (não pode ser vazio), Modelo do Veículo (não pode ser vazio), Ano de fabricação (deve ser entre 1930 até ano atual(2024) e tipo int) e Placa do veículo (será convertida para maiúsculo))

            fabricante\_veiculo = input("Digite a fabricante do seu veículo: ")

            while fabricante\_veiculo == "":

                print("Erro! Marca do veículo não pode estar vazia.")

                fabricante\_veiculo = input("Digite a marca do seu veículo: ")

            modelo\_veiculo = input("Digite o modelo do seu veículo: ")

            while modelo\_veiculo == "":

                print("Erro! Marca do veículo não pode estar vazia.")

                modelo\_veiculo = input("Digite a modelo do seu veículo: ")

            ano\_veiculo = int(input("Ano de fabricação: "))

            while ano\_veiculo == "" or ano\_veiculo < 1930 or ano\_veiculo > 2024:

                print("ERRO. Digite um ano válido")

                ano\_veiculo = int(input("Ano de fabricação: "))

            placa\_veiculo = input("Placa do veículo: ").upper()

            while placa\_veiculo == "":

                print(f"ERRO. Placa {placa\_veiculo} digitada é inválida. Digite novamente.")

                placa\_veiculo = input("Placa do veículo: ").upper()

            os.system('cls')

            print(f'{fabricante\_veiculo} || {modelo\_veiculo} || {ano\_veiculo} || {placa\_veiculo}') #Mostra os dados coletados para verificação

        case 2: # Lista opções de problemas comuns automotivos, usuário escolhe a opção com números e recebe uma descrição do problema

            try:

                problema\_painel\_escolhido = int(input("""

1 - Óleo

2 - Bateria

3 - Radiador

Opção: """))

                match problema\_painel\_escolhido: #Identifica a opção escolhida e retorna a descrição correspondente // COMO FAZER O PROGRAMA RETORNAR PARA MENU DE ESCOLHA

                    case 1:

                        print("Descrição de problemas e soluções relacionadas ao óleo")

                    case 2:

                        print("Descrição de problemas e soluções relacionadas a bateria")

                    case 3:

                        print("Descrição de problemas e soluções relacionadas ao radiador")

                    case \_:

                        print("Opção inválida! digite o número correspondente com a opção desejada.")

            except:

                print("Opção inválida! digite o número correspondente com a opção desejada.")

        case 3:

            check\_pneu = input("Checou o estado do Pneu? S/N: ").upper()

            while check\_pneu != "S" and check\_pneu != "N":

                print(f"ERRO! Tecla '{check\_pneu}' inválida. Digite S ou N")

                check\_pneu = input("Checou o estado do Pneu? S/N: ").upper()

            check\_calibrar = input("Calibrou o Pneu? S/N: ").upper()

            while check\_calibrar != "S" and check\_calibrar != "N":

                print(f"ERRO! Tecla '{check\_calibrar}' inválida. Digite S ou N")

                check\_calibrar = input("Checou o estado do Pneu? S/N: ").upper()

            check\_oleo = input("Checou o nível do Óleo? S/N: ").upper()

            while check\_oleo != "S" and check\_oleo != "N":

                print(f"ERRO! Tecla '{check\_oleo}' inválida. Digite S ou N")

                check\_oleo = input("Checou o estado do Pneu? S/N: ").upper()

            check\_farois = input("Checou as Luzes do carro? S/N: ").upper()

            while check\_farois != "S" and check\_farois != "N":

                print(f"ERRO! Tecla '{check\_farois}' inválida. Digite S ou N")

                check\_farois = input("Checou o estado do Pneu? S/N: ").upper()

            os.system('cls')

            print(f"""

CheckList Diário de Bordo:

Pneu checado:...................... {check\_pneu}

Pneu calibrado:.................... {check\_calibrar}

Checou o óleo:..................... {check\_oleo}

Checou as luzes:................... {check\_farois}

""")

        case 4:

            os.system('cls')

        case \_:

            os.system('cls')

            print("Opção inválida! digite o número correspondente com a opção desejada.")

except:

    os.system('cls')

    print("Opção inválida! digite o número correspondente com a opção desejada.")