

FIAP

The background of the cover features the Python logo, which consists of two interlocking snakes. The left snake is blue and the right snake is yellow. The word "ChatMech" is written in a large, bold, black sans-serif font, centered over the logo.

ChatMech

COMPUTATIONAL THINKING USING PYTHON

João Victor Michaeli de Bem - 555678

Gabriel Diaz Menezes - 555019

Sumário

1 - Introdução.....	3
2 - Mudanças.....	4
3 - Descritivo.....	5
3 - Código-fonte.....	8

Introdução

Foi apresentado pela Porto Seguro um desafio no qual devemos criar um projeto que atenda o usuário via aplicação, como se fosse uma mecânica online. A principal ferramenta será a utilização de um ChatBot, o qual, por meio de uma conversa, permitirá ao usuário descrever o problema enfrentado em seu veículo, fornecendo informações como marca, modelo, ano e quilometragem. Com esses dados, o bot será capaz de gerar um breve diagnóstico do problema e a respectiva solução para a manutenção do veículo, além de apresentar um possível orçamento das peças problemáticas e listar todas as oficinas que oferecem o serviço solicitado, indicando também as vagas disponíveis para o atendimento, a fim de facilitar e agilizar a ida à mecânica. Dessa forma, o cliente chegará à oficina já com um diagnóstico provável do problema, e a própria mecânica estará ciente da situação, uma vez que também terá recebido o diagnóstico por meio do site.

Com essa abordagem em mente, desenvolvemos em Python um código que contempla um menu com algumas opções que estarão presentes em nosso site.

Mudanças

Na primeira entrega, utilizamos apenas o match case e estruturas condicionais e de repetição. No entanto, isso resultou em alguns loops infinitos. Por isso, decidimos reformular quase todo o código. Removemos o match case e introduzimos funções que contêm as estruturas implementadas, resolvendo assim o problema da primeira entrega.

Descritivo

- `exibir_menu`: Nessa função, deixamos apenas o menu principal, que contém suas opções e a opção de retornar à tela de login. Inicialmente, algumas funções ainda não estão presentes, pois fazem parte de outras disciplinas, como a implementação do ChatBot em si. No entanto, as outras opções são funcionais. A primeira opção é um MVP, que é o cadastramento do veículo. Quando acessada, direciona para outra função, onde serão solicitadas as informações do veículo. Também há a opção de agendamento, onde serão solicitados o dia, mês e hora do serviço.
- `agendar_servico`: Função para fazer o agendamento do serviço, onde serão solicitadas informações como dia, mês e hora. Há uma verificação dos valores informados e, em seguida, o agendamento é concluído.
- `acessar_usuario`: A primeira tela apresentada após o início do programa. Nela, será solicitado o registro do usuário, chamando uma função destinada ao cadastramento. Há também a opção de acessar a conversa com o ChatBot sem o cadastramento, embora essa funcionalidade só será implementada posteriormente com a integração do ChatBot. As outras opções incluem fazer login e encerrar o programa.

- `registrar_usuario`: Função para o registro do usuário, onde uma lista vazia é utilizada para armazenar as informações de login e senha. Usamos uma lógica simples, adicionando o login e a senha à lista com o método `append`.
- `verificar_login`: Função para acessar o programa, continuando o processo de registro. Ela verifica a lista onde estão armazenados o login e a senha. Para acessar, é necessário utilizar o mesmo login e senha cadastrados anteriormente. Se as informações estiverem incorretas, o acesso será negado. Como ainda não há implementação de um banco de dados para armazenar múltiplos registros, cada vez que o programa é iniciado, o usuário precisa se registrar novamente, pois a lista estará vazia.
- `cadastros`: Função para cadastrar o veículo do usuário. Essa função é acessada a partir do menu principal, escolhendo a opção de cadastro. Ela oferece as opções de cadastrar veículo, mostrar veículos cadastrados e voltar ao menu inicial. Há uma lista vazia chamada `veículos`, onde são armazenadas as informações fornecidas pelo usuário. A opção de mostrar veículos cadastrados retorna os veículos registrados na primeira opção. Como não há um banco de dados, só é possível armazenar um veículo por vez.

- `cadastrar_veiculo`: Esta função é destinada ao cadastro de veículos. Ao ser acessada, solicita informações como placa, modelo e dono do veículo, armazenando esses dados em uma lista. O cadastro com as informações do veículo será retornado caso o usuário selecione a opção de mostrar os veículos cadastrados.

Código Fonte

```
1 import os
2 os.system("cls")
3
4 #----- Funções
5 class ExitProgram(Exception):
6     pass
7
8 def exibir_menu():
9     while True:
10         print("\nBem vindo ao ChatMech\n")
11         "1 - Cadastrar um veículo\n"
12         "2 - Conversar com o Mechinho\n"
13         "3 - Meus orçamentos\n"
14         "4 - Histórico de manutenções\n"
15         "5 - Agendar serviços\n"
16         "6 - Quem somos\n"
17         "7 - Retornar a tela de login\n")
18
19     try:
20         opcao = int(input("Escolha uma opção: "))
21
22         if opcao == 1:
23             os.system("cls")
24             cadastros()
25
26         elif opcao == 2:
27             print("\nBem vindo ao ChatMech, nosso mecânico virtual, qual o seu problema?\n") # Aqui o
28             # programa irá retornar ao menu, pois só irá ter continuidade com a implementação do ChatBot
29
30         elif opcao == 3:
31             print("\nOrçamentos das peças e serviço relacionado ao diagnóstico de seu veículo:\n") #
32             # Aqui o programa irá retornar ao menu, pois só irá ter continuidade com a implementação do ChatBot
33
34         elif opcao == 4:
35             print("\nHistórico de serviços feitos pelo o Mechinho:\n") # Aqui o programa irá se
36             # retornar ao menu, pois só irá ter continuidade com a implementação do ChatBot
37
38         elif opcao == 5:
39             d = int(input("\nQual dia gostaria de agendar seu serviço?"))
40             m = int(input("\nQual o mês?"))
41             h = int(input("\nQual o horário do agendamento?"))
42             return agendar_servico(dia=d, mes=m, hora=h)
43
44         elif opcao == 6:
45             print("\nSomos uma solução rápida e prática para problemas mecânicos em geral, atendemos
46             via internet por meio do Mechinho, nosso ChatBot, onde ele fará uma série de perguntas
47             sobre o problema de seu veículo e por meio de sua inteligência ele será capaz de dar um
48             diagnóstico da possível solução e em quais mecânicas atendem o caso, além de dar um breve
49             orçamento das peças necessárias para a manutenção, o preço é obtido por meio do mercado
50             geral, podendo ter mudanças por região.")
51
52         elif opcao == 7:
53             print("Saindo...")
54             break
55         else:
56             print("Opção inválida. Tente novamente.")
57
58     except ValueError:
59         print("Entrada inválida Por favor, digite um número.")
60         os.system("cls")
61
62 def agendar_servico(dia, mes, hora):
63     # Verificação se os valores de dia e mês são números
64     dia = int(dia)
65     mes = int(mes)
66
67     # Assume o formato de hora como 'hh:mm'
68     hora_parts = hora.split(':')
69     hora_valid = len(hora_parts) == 2 and all(part.isdigit() for part in hora_parts)
70
71     if not (1 <= dia <= 31):
72         raise ValueError("Valor de dia inválido. Use um valor entre 1 e 31.")
73     if not (1 <= mes <= 12):
74         raise ValueError("Valor de mês inválido. Use um valor entre 1 e 12.")
75     if not hora_valid:
76         raise ValueError("Formato de hora inválido. Use 'hh:mm'.")
77
78     # Conversão para inteiros para validações adicionais
79     hora_int = list(map(int, hora_parts))
80
81     # Verificação de valores de hora
82     if not (0 <= hora_int[0] < 24 and 0 <= hora_int[1] < 60):
83         raise ValueError("Valores de hora ou minuto inválidos.")
84
85     # Se todas as validações passaram, retorna um agendamento bem-sucedido
86     print(f"Serviço agendado para o dia {dia}/{mes} às {hora}.")
87     return {'dia': dia, 'mes': mes, 'hora': hora}
```



```

79 def acessar_usuario():
80     try:
81         while True:
82             print("\n1 - Registrar usuário\n"
83                   "2 - Fazer login\n"
84                   "3 - Encerrar programa\n")
85             try:
86                 opcao = int(input("Escolha uma opção: "))
87                 if opcao == 1: # Irá chamar a tela de registro de usuário
88                     os.system("cls")
89                     registrar_usuario()
90                 elif opcao == 2: # Irá chamar a tela de login
91                     os.system("cls")
92                     if verificar_login():
93                         print("Acesso concedido")
94
95                     else:
96                         os.system("cls")
97                         print("Acesso negado, você não possui uma conta registrada...\n"
98                               "Por favor, Registre-se aqui: \n")
99                         return registrar_usuario()
100                 elif opcao == 3: # Sairá do programa
101                     print("Saindo...")
102                     raise ExitProgram
103                 else:
104                     print("Opção inválida. Tente novamente.\n")
105             except ValueError:
106                 print("Entrada inválida! Por favor, digite um número.")
107                 os.system("cls")
108         except ExitProgram:
109             print("Programa encerrado.")
110
111 usuarios = [] # Uma lista varia que armazena os dados de login e senha
112
113 def registrar_usuario(): # Tela de registro
114     os.system("cls")
115     print("Bem vindo, Cadastre-se em nosso sistema: \n")
116     login = input("Digite seu login: ")
117     senha = input("Digite sua senha: ")
118     usuarios.append({'login': login, 'senha': senha}) # Adicionará o login e senha na lista
119     print("\nUsuário registrado com sucesso!\n")
120     return verificar_login()
121
122 def verificar_login(): # Tela de verificação
123     print("Para realizar seu login, preencha as informações abaixo:\n")
124     login = input("Digite seu login: ")
125     senha = input("Digite sua senha: ")
126
127     for usuario in usuarios: # Aqui o usuário terá que digitar o mesmo login e senha que foi digitado na
128                             # tela de registro, pois é a informação que está armazenada na lista, caso digite outra informação,
129                             # resultará em dados incorretos, e ficará dando loop até digitar as informações dadas no registro
130         if usuario['login'] == login and usuario['senha'] == senha:
131             print("Login bem-sucedido!")
132             os.system("cls")
133             return exibir_menu()
134         else:
135             print("\nLogin ou senha incorretos.\n")
136             return verificar_login()
137
138 veiculos = [] # Uma lista varia que armazena os dados do veículo

```

```

138 def cadastrar():
139     while True:
140         print("1 - Cadastrar veículo\n"
141             "2 - Mostrar veículos cadastrados\n"
142             "3 - Voltar ao Menu inicial\n")
143
144         opcao = int(input("Escolha uma opção: "))
145
146         if opcao == 1:
147             os.system("cls")
148             cadastrar_veiculo()
149         elif opcao == 2:
150             if veiculos:
151                 print("\nVeículos cadastrados:\n")
152                 for veiculo in veiculos:
153                     print(f"Placa: {veiculo['placa']}\nModelo: {veiculo['modelo']}\nDono: {veiculo['dono']}\n")
154             else:
155                 print("\nNenhum veículo cadastrado.\n")
156         elif opcao == 3:
157             print("Saindo...")
158             os.system("cls")
159             break
160         else:
161             print("Opção inválida. Tente novamente.")
162
163
164
165 def cadastrar_veiculo():
166     placa = input("\nDigite a placa do seu veículo: ")
167     modelo = input("\nDigite o modelo do seu veículo: ")
168     dono = input("\nDigite o nome do dono do veículo: ")
169
170     # Criar um dicionário com os dados do veículo
171     veiculo = {
172         'placa': placa,
173         'modelo': modelo,
174         'dono': dono
175     }
176
177     # Adicionar o dicionário a lista de veículos
178     veiculos.append(veiculo)
179     print("\nVeículo cadastrado com sucesso!\n")
180
181 #----- Programa principal
182
183 os.system("cls")
184
185 print("Bem vindo ao ChatMech, acesse sua conta: \n")
186 print("1 - Acessar ou Cadastrar usuário"
187     "\n2 - Acessar o Mechzinho sem cadastro")
188
189 opcao = int(input("\nEscolha uma opção: "))
190
191 match opcao:
192     case 1:
193         print("\nO que você deseja ?")
194         acessar_usuario()
195     case 2:
196         print("Bem vindo ao ChatMech, nosso mecânico virtual, qual o seu problema ?") #Aqui será a
            implementação com o ChatBot, então como não é possível ainda, a opção não retorna nada

```