



Lordes do Tempo

TRABALHO DE PROGRAMAÇÃO



Funções principais

- O programa fornece dados climáticos de qualquer lugar, desde que o mesmo tenha um estado e o nome da cidade
- Velocidade do vento Km/h
- Umidade
- Temperatura média/máxima/mínima
- Pressão

Alguns Imports iniciais

- Foi importado telebot e telepot
- Do telebot importamos « types »
- Importamos também as funções criadas

```
progprin.py  funcoes.py
1  import telebot
2  import telepot
3  from telebot import types
4  from funcoes import *
5
```

```
6
7 bot_token= "626239086:AAGtqcMKcV-DRd6jBfh_lGoXU_Q_zVut2j8"
8
9 t=telebot.TeleBot(bot_token)
10
```

- Código do bot que foi usado

```
11 opcoes=["ClimatempoAPI","OpenWeatherAPI"]
12 selecionarOpcao = types.ReplyKeyboardMarkup()
13 selecionarOpcao.add(*opcoes)
14
```

- Aqui o usuário escolhe entre duas opções

Mensagens iniciais do Bot

```
23 @bot.message_handler(commands=['start'])
24 def boas_vindas(message):
25     usu_id=message.chat.id
26     nome_usu=message.chat.first_name
27     bot.send_message(usu_id,"Bem-vindo(a) {0} ao AccuWeatherBot".format(nome_usu))
28     bot.send_message(usu_id,"Este bot fornece os dados meteorológicos momentâneos do local escolhido")
29     bot.send_message(usu_id,"Escolha qual API você deseja ",reply_markup=selecionarOpcao)
30     all_messages.append(message.text)
31
```

- Função boas_vindas
- O bot reproduz algumas mensagens de como ele funciona e para o que ele serve

Solicitando as informações

```
31
32
33 @bot.message_handler(func=lambda message: message.text == opcoes[0] or message.text == opcoes[1])
34 def local1(message):
35     usu_id=message.chat.id
36     bot.send_message(usu_id,"Digite o nome do estado, por favor em siglas (ex: PR)", reply_markup=cancelar)
37     all_messages.append(message.text)
38
39 @bot.message_handler(func=lambda message: message.text == "Cancelar")
40 def correcao(message):
41     usu_id=message.chat.id
42     bot.send_message(usu_id,"Escolha outra opção",reply_markup=selecionarOpcao)
43     all_messages.append(message.text)
44
45
```

- O bot solicita alguns dados, como por exemplo a sigla do estado
- Caso o usuário digite errado, o bot não irá funcionar

Solicitando informação

```
44
45
46 @bot.message_handler(func=lambda message: message.text.isupper())
47 def continuacao(message):
48     usu_id=message.chat.id
49     bot.send_message(usu_id,"Estado escolhido",reply_markup=esconderTeclado)
50     bot.send_message(usu_id,"Digite o nome da cidade, (ex: curitiba)")
51     all_messages.append(message.text)
52
```

- Depois de solicitar o estado em siglas, o bot irá pedir o nome da cidade

Informações

```
52
53 @bot.message_handler(func=lambda message: message.text.islower())
54 def final(message):
55     usu_id=message.chat.id
56     all_messages.append(message.text)
57     if all_messages[-3] == "OpenWeatherAPI":
58         cid_info = all_messages[-1]
59         cid, country, humidity, pressure, temp_min, temp_max, temp, wind_speed, condition = dados_momento_info_open(cid_info)
60         bot.send_message(usu_id,("Seu país: {0}".format(country)))
61         bot.send_message(usu_id,("Sua cidade: {0}".format(cid)))
62         bot.send_message(usu_id,("Temperatura momentânea: {0:.2f}°C".format(temp-273.15)))
63         bot.send_message(usu_id,("Temperatura Máxima: {0:.2f}°C".format(temp_max-273.15)))
64         bot.send_message(usu_id,("Temperatura Mínima: {0:.2f}°C".format(temp_min-273.15)))
65         bot.send_message(usu_id,("Condição: {0}".format(condition)))
66         bot.send_message(usu_id,("Pressão: {0} hPa".format(pressure)))
67         bot.send_message(usu_id,("Velocidade do vento: {0} km/h".format(wind_speed*3.6)))
68         bot.send_message(usu_id,("Umidade relativa(%): {0}".format(humidity)))
```

- Se o usuário escolheu a opção “OpenWeatherAPI”
- Temperatura transformada em °C
- Velocidade do vento em Km/h (*)
- O bot pega os dados do site e repassa via telegram ao usuário

Pegando os dados

```
import requests
import json

def localizacao(cid,state):
    cidade = cid
    estado = state
    requisicao = requests.get("http://apiadvisor.climatempo.com.br/api/v1/locale/city?name="+cidade+"&state="+estado+"&token=cb687205e4827654ae34805feff813e1")
    loc=json.loads(requisicao.text)
    lista_loc=list(loc)
    id_cid=str(lista_loc[0]['id'])
    name_cid=lista_loc[0]['name']
    name_state=lista_loc[0]['state']
    name_country=lista_loc[0]['country']
    return name_cid,name_state,id_cid,name_country
```

- Criamos uma função para retornar os dados de localização que o usuário deseja

```
69 elif all_messages[-3] == "ClimatempoAPI":
70     cid = all_messages[-1]
71     state = all_messages[-2]
72     temp, wind_velocity, wind_direction, humidity, condition, pressure, sensation, date, name_cid, name_state, id_cid, name_country = dados_momento_info_clima(cid, state)
73     bot.send_message(usu_id, ("Seu país: {}".format(name_country)))
74     bot.send_message(usu_id, ("Seu estado: {}".format(name_state)))
75     bot.send_message(usu_id, ("Sua cidade: {}".format(name_cid)))
76     bot.send_message(usu_id, ("Data e horário: {}".format(date)))
77     bot.send_message(usu_id, ("Temperatura de hoje: {}".format(temp)))
78     bot.send_message(usu_id, ("Sensação térmica: {}".format(sensation)))
79     bot.send_message(usu_id, ("Condição: {}".format(condition)))
80     bot.send_message(usu_id, ("Pressão: {} hPa".format(pressure)))
81     bot.send_message(usu_id, ("Velocidade do vento: {} km/h".format(wind_velocity)))
82     bot.send_message(usu_id, ("Direção do vento: {}".format(wind_direction)))
83     bot.send_message(usu_id, ("Umidade relativa(%): {}".format(humidity)))
84
85
86
87 bot.polling()
```

- Se o usuário escolheu a opção “ClimaTempoAPI”
- O bot envia os dados via telegram ao usuário

Fornecendo os dados

```
15
16 def dados_momento_info_clima(cid,state):
17     name_cid,name_state,id_cid,name_country = localizacao(cid,state)
18     id_cid= id_cid
19     requisicao = requests.get("http://apiadvisor.climatempo.com.br/api/v1/weather/locale/"+id_cid+"/current?token=cb687205e4827654ae34805feff813e1")
20     lista_prev=json.loads(requisicao.text)
21     temp=lista_prev['data']['temperature']
22     wind_direction=lista_prev['data']['wind_direction']
23     wind_velocity=lista_prev['data']['wind_velocity']
24     humidity=lista_prev['data']['humidity']
25     condition=lista_prev['data']['condition']
26     pressure=lista_prev['data']['pressure']
27     sensation=lista_prev['data']['sensation']
28     date=lista_prev['data']['date']
29     return temp,wind_velocity,wind_direction,humidity,condition,pressure,sensation,date,name_cid,name_state,id_cid,name_country
30
```

- A função seleciona os dados do site climatempo

- Elias Makoto
- Marcos Paulo
- Renato Hopfer

REFERÊNCIAS

- <https://github.com/jacksonpradolima/ufpr-ci182-trabalhos/blob/master/201802/Matematica/CI182MAT1/6%20-%20Eru/relatorio.pdf>
- https://github.com/jacksonpradolima/ufpr-ci182-trabalhos/blob/master/201802/Matematica/CI182MAT1/6%20-%20Eru/codigo/c%C3%B3digo_comentado.py#L203
- <https://www.youtube.com/watch?v=jhFsFZXZbu4&t=169s>
- <https://core.telegram.org/bots/api#replykeyboardmarkup>
- <https://core.telegram.org/bots#keyboards>
- <https://pypi.org/project/pyTelegramBotAPI/>
- <https://github.com/adinan-cenci/climatempo-api>
- <https://www.freecodecamp.org/news/learn-to-build-your-first-bot-in-telegram-with-python-4c99526765e4/>
- <https://github.com/diegosmaia/Prev-tempo-climatempo/blob/master/climatempo.py>
- <https://github.com/diegosmaia/Prev-tempo-climatempo/blob/master/tempagora-antigo.py>
- <http://apiadvisor.climatempo.com.br/doc/index.html#api-Locale-GetCityByNameAndState>
- <https://imasters.com.br/back-end/aprendendo-sobre-web-scraping-em-python-utilizando-beautifulsoup>
- https://www.youtube.com/watch?v=a-M7_QxQ2kA