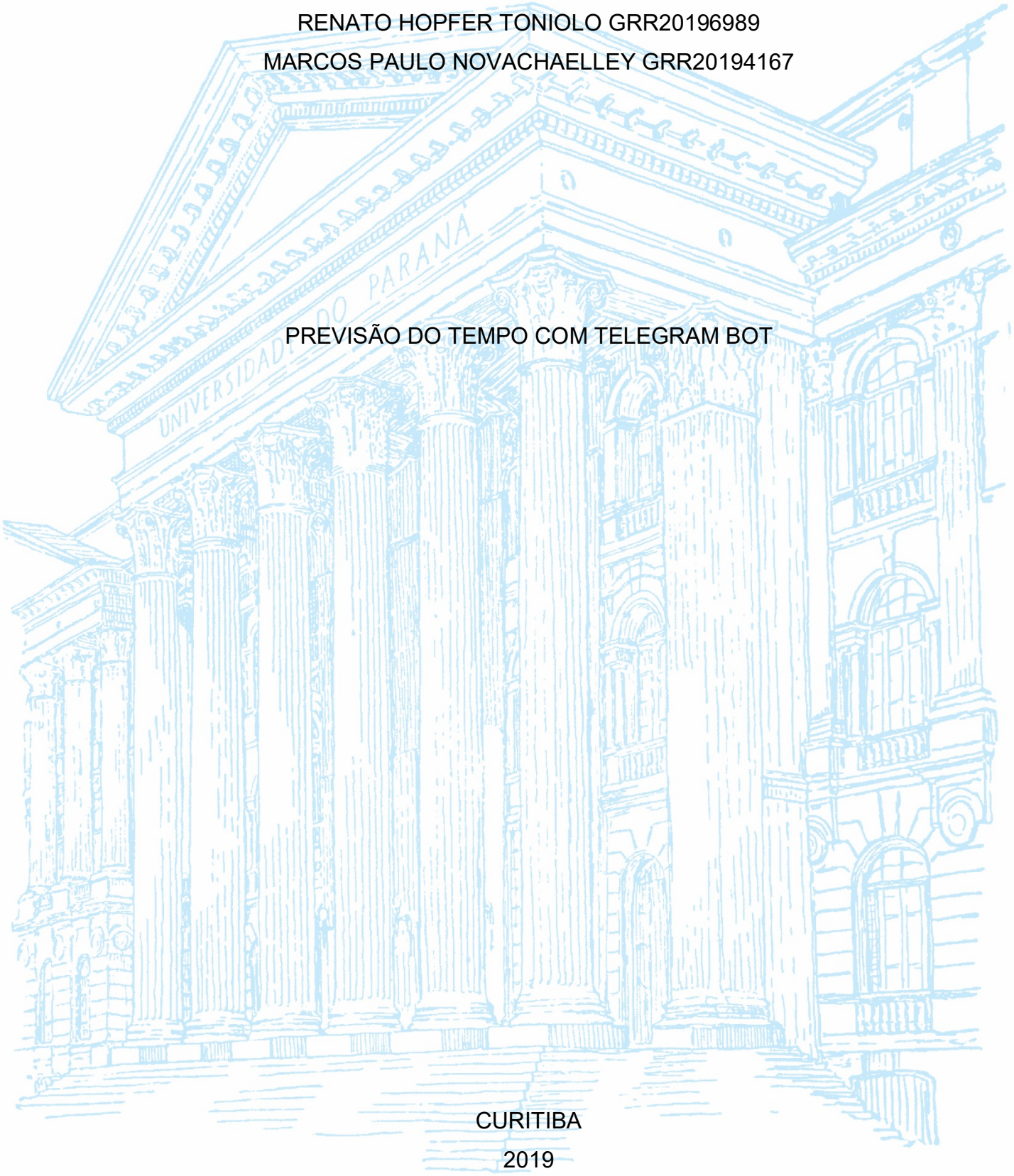
UNIVERSIDADE FEDERAL DO PARANÁ

ELIAS MAKOTO GRR20194262
RENATO HOPFER TONIOLO GRR20196989
MARCOS PAULO NOVACHAELLEY GRR20194167



PREVISÃO DO TEMPO COM TELEGRAM BOT

CURITIBA
2019

RESUMO

Este trabalho refere-se a criação de um bot para o Telegram, com uso da linguagem Python, bot este voltado para informação do tempo nacional e internacionalmente. Serão apresentadas o código, as funções criadas para o correto funcionamento do programa, bem como resultados e dificuldades encontradas.

Palavras-chave: Bot, Telegram, clima, tempo.

LISTA DE FIGURAS

SUMÁRIO

1 INTRODUÇÃO

Devido ao contexto extremamente conectado em que vivemos, a informação se tornou algo praticamente ilimitado, computadores gigantescos passaram a ser mais portáteis e com a invenção de celulares potentes, as fronteiras foram praticamente extintas. Nesse contexto muitas tecnologias surgiram, como futuros agrônomos pensando na comodidade e praticidade, criamos o AccuWeatherBot, visando ser um robô na plataforma Telegram voltado para informação de tempo, isso tudo em um aplicativo de chat, atualmento o tipo de app mais utilizado mundialmente.

2 BIBLIOTECAS UTILIZADAS

Foram usadas Telepot e Telebot, que são bibliotecas do Python que te permite se conectar à API de Bots do Telegram. Além de Requests, uma biblioteca que analisa dados via HTTP; e JSON, que pode operar com objetos json originários de arquivos ou strings, ao decodificar o objeto, a biblioteca o converte para listas ou dicionários.

3 FUNÇÕES CRIADAS E IMPLEMENTAÇÃO

```
def boas_vindas(message):
    usu_id=message.chat.id
    nome_usu=message.chat.first_name
    bot.send_message(usu_id,"Bem-vindo(a) {0} ao AccuWeatherBot".format(nome_usu))
    bot.send_message(usu_id,"Este bot fornece os dados meteorológicos momentâneos do local escolhido")
    bot.send_message(usu_id,"Escolha qual API você deseja ",reply_markup=selecionarOpcao)
    all_messages.append(message.text)
```

Função da boas vindas ao usuário quando ele digita /start. (Figura. 1)

```
def local1(message):
    usu_id=message.chat.id
    bot.send_message(usu_id,"Digite o nome do estado, por favor em siglas (ex: PR)", reply_markup=cancelar)
    all_messages.append(message.text)
```

É requisitado do usuário a sigla do estado, caso o usuário queira cancelar a opção antes escolhida, é fornecido a opção cancelar. (Figura. 2)

```
def correcao(message):
    usu_id=message.chat.id
    bot.send_message(usu_id,"Escolha outra opção",reply_markup=selecionarOpcao)
    all_messages.append(message.text)
```

Essa função irá ocorrer quando o usuário optar por cancelar. Será requisitado que escolha outra opção. (Figura. 3)

```
def continuacao(message):
    usu_id=message.chat.id
    bot.send_message(usu_id,"Estado escolhido",reply_markup=esconderTeclado)
    bot.send_message(usu_id,"Digite o nome da cidade, (ex: curitiba)")
    all_messages.append(message.text)
```

Informa ao usuário o Estado de sua escolha e solicita que seja digitado o nome da cidade. (Figura. 4)

```python
def final(message):
    usu_id=message.chat.id
    all_messages.append(message.text)
    if all_messages[-3] == "OpenWeatherAPI":
        cid_info = all_messages[-1]
        cid, country, humidity, pressure, temp_min, temp_max, temp, wind_speed, condition = dados_momento_info_open(cid_info)
        bot.send_message(usu_id,("Seu país: {0}".format(country)))
        bot.send_message(usu_id,("Sua cidade: {0}".format(cid)))
        bot.send_message(usu_id,("Temperatura momentânea: {0:.2f}°C".format(temp-273.15)))
        bot.send_message(usu_id,("Temperatura Máxima: {0:.2f}°C".format(temp_max-273.15)))
        bot.send_message(usu_id,("Temperatura Mínima: {0:.2f}°C".format(temp_min-273.15)))
        bot.send_message(usu_id,("Condição: {0}".format(condition)))
        bot.send_message(usu_id,("Pressão: {0} hPa".format(pressure)))
        bot.send_message(usu_id,("Velocidade do vento: {0} km/h".format(wind_speed*3.6)))
        bot.send_message(usu_id,("Umidade relativa(%): {0}".format(humidity)))
    elif all_messages[-3] == "ClimatempoAPI":
        cid = all_messages[-1]
        state = all_messages[-2]
        temp,wind_velocity,wind_direction,humidity,condition,pressure,sensation,date,name_cid,name_state,id_cid,name_country = dados_momento_info_clima(cid,state)
        bot.send_message(usu_id,("Seu país: {0}".format(name_country)))
        bot.send_message(usu_id,("Seu estado: {0}".format(name_state)))
        bot.send_message(usu_id,("Sua cidade: {0}".format(name_cid)))
        bot.send_message(usu_id,("Data e horário: {0}".format(date)))
        bot.send_message(usu_id,("Temperatura de hoje: {0}°C".format(temp)))
        bot.send_message(usu_id,("Sensação térmica: {0}°C".format(sensation)))
        bot.send_message(usu_id,("Condição: {0}".format(condition)))
        bot.send_message(usu_id,("Pressão: {0} hPa".format(pressure)))
        bot.send_message(usu_id,("Velocidade do vento: {0} km/h".format(wind_velocity)))
        bot.send_message(usu_id,("Direção do vento: {0}".format(wind_direction)))
        bot.send_message(usu_id,("Umidade relativa(%): {0}".format(humidity)))
```

Nessa função o dados serão obtidos a partir da lista all_messages, não importa quantas informações o usuário adicione a lista, os dados API, estado e cidade sempre irão corresponder as posições [-3],[-2] e [-1] respectivamente. Os dados obtidos pelas funções criadas, serão enviadas ao usuário. (Figura. 5)

```python
def localizacao(cid,state):
    cidade = cid
    estado = state
    requisição = requests.get("http://apiadvisor.climatempo.com.br/api/v1/locale/city?name="+cidade+"&state="+estado+"&token=cb687205e4827654ae34805feff813e1")
    loc=json.loads(requisição.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
```

Função que fornece o nome da cidade, do estado, do país e o id da cidade, os dados fornecidos se encontram na forma de listas e sublistas. (Figura. 6)

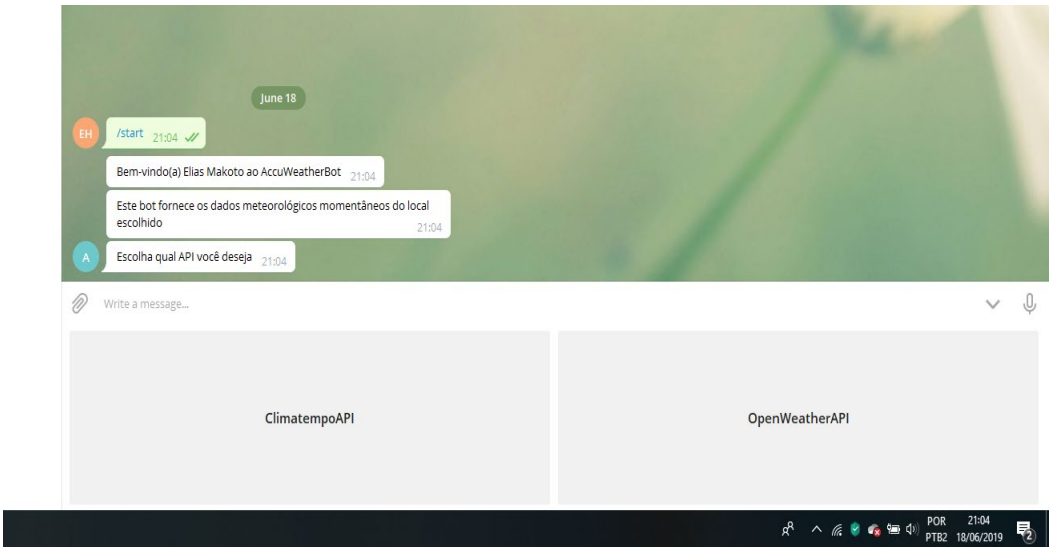
```python
def dados_momento_info_clima(cid,state):
    name_cid,name_state,id_cid,name_country = localizacao(cid,state)
    id_cid= id_cid
    requisição = requests.get("http://apiadvisor.climatempo.com.br/api/v1/weather/locale/"+id_cid+"/current?token=cb687205e4827654ae34805feff813e1")
    lista_prev=json.loads(requisição.text)
    temp=lista_prev['data']['temperature']
    wind_direction=lista_prev['data']['wind_direction']
    wind_velocity=lista_prev['data']['wind_velocity']
    humidity=lista_prev['data']['humidity']
    condition=lista_prev['data']['condition']
    pressure=lista_prev['data']['pressure']
    sensation=lista_prev['data']['sensation']
    date=lista_prev['data']['date']
    return temp,wind_velocity,wind_direction,humidity,condition,pressure,sensation,date,name_cid,name_state,id_cid,name_country
```

Com o id obtido pela função localização, ocorre a obtenção dos dados meteorológicos com o API do Climatempo, os dados fornecidos se encontram na forma de listas e sublistas. (Figura. 7)

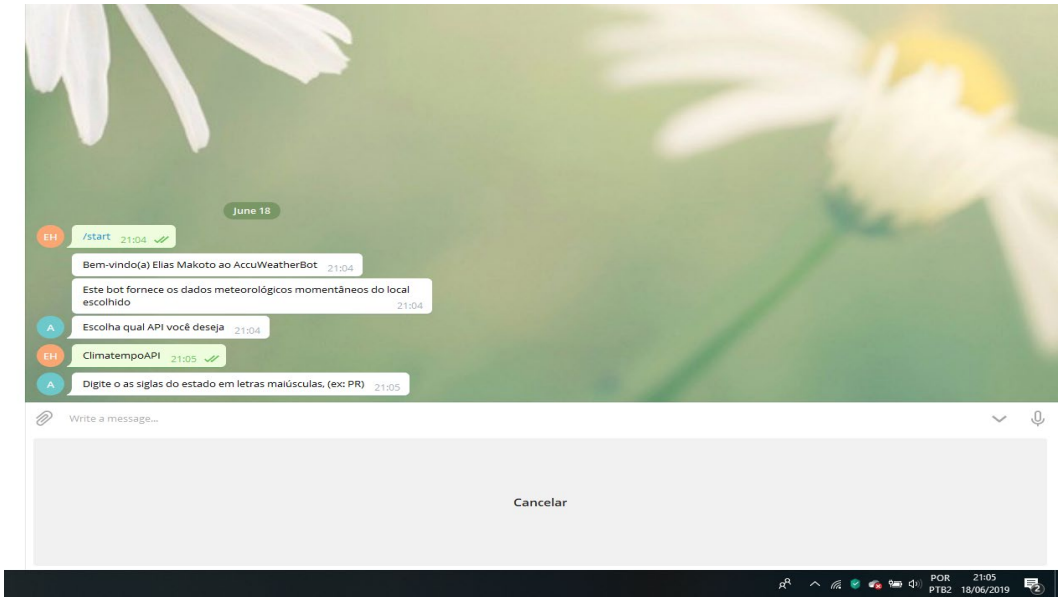
```python
def dados_momento_info_open(cid_info):
    cid_info = cid_info
    requisição = requests.get('https://api.openweathermap.org/data/2.5/weather?q='+cid_info+'&appid=032724f464f443a20fa48ca83b587ecd')
    lista_prev=json.loads(requisição.text)
    condition = lista_prev['weather'][0]['description']
    temp = lista_prev['main']['temp']
    temp_max = lista_prev['main']['temp_max']
    temp_min = lista_prev['main']['temp_min']
    pressure = lista_prev['main']['pressure']
    humidity = lista_prev['main']['humidity']
    wind_speed = lista_prev['wind']['speed']
    country = lista_prev['sys']['country']
    cid = lista_prev['name']
    return cid, country, humidity, pressure, temp_min, temp_max, temp, wind_speed, condition
```

Cid_info corresponde a cidade que o usuário digitou, que corresponderá a posição [-1] da lista all_messages. Serão obtidos os dados meteorológicos usando o API do OpenWeather, os dados fornecidos se encontram na forma de listas e sublistas. (Figura. 8)

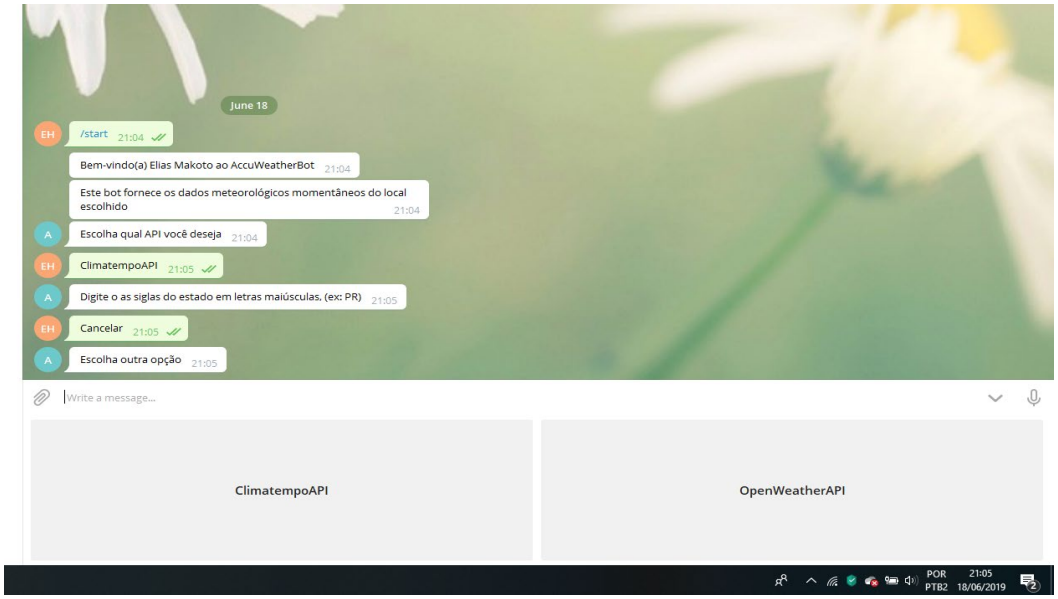
4 RESULTADOS DA IMPLEMENTAÇÃO



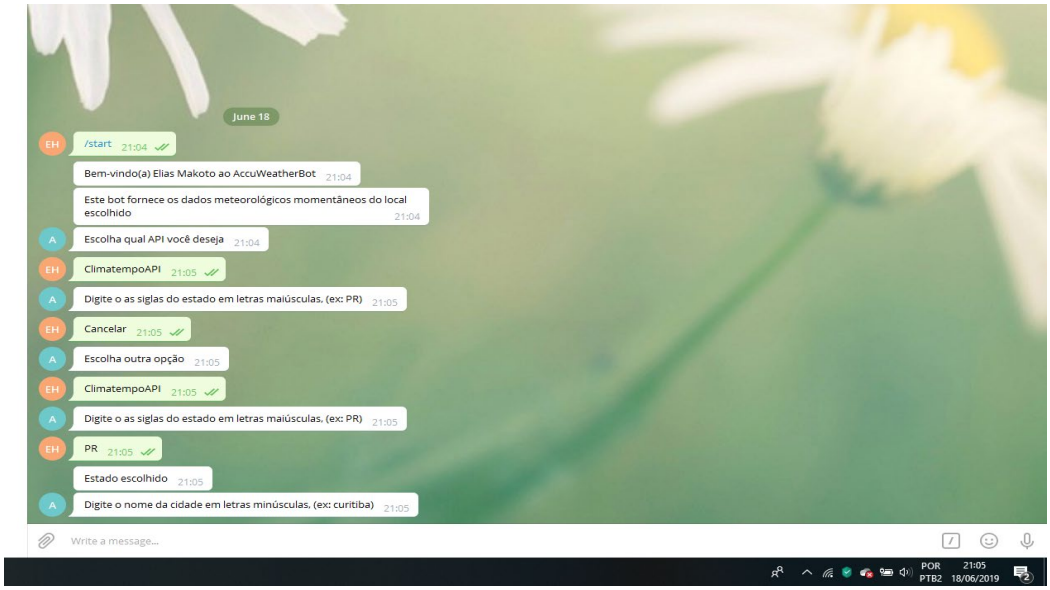
Inicialmente o bot irá lhe dar boas vindas e solicitar qual API você deseja, Clima Tempo ou OpenWeather. (Figura. 9)



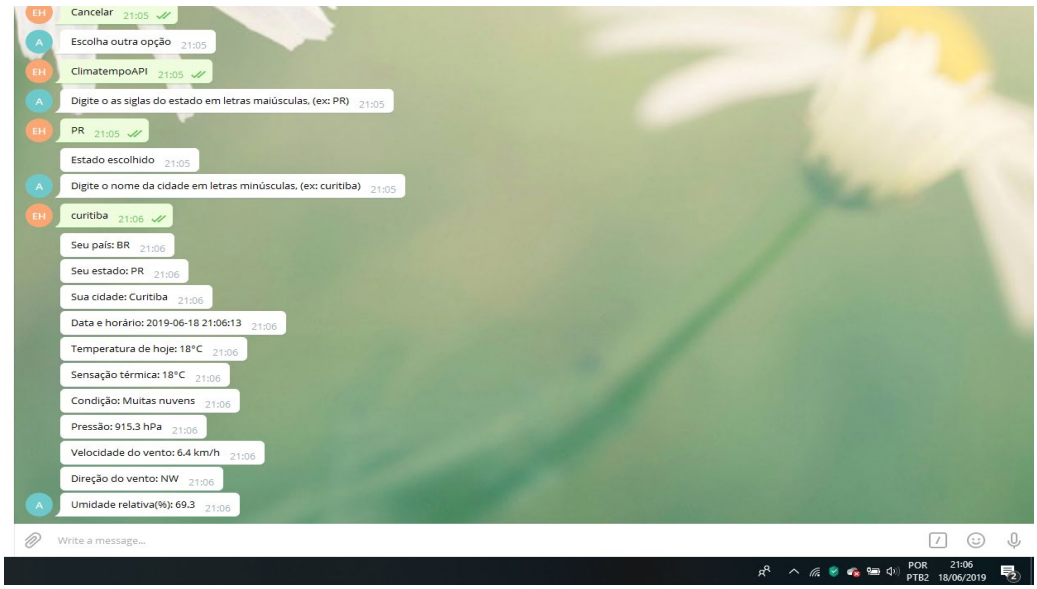
Ele solicitará o estado que o usuário deseja (em letras maiúsculas), e dará a opção de cancelamento, se assim desejar o usuário. (Figura. 10)



Caso cancelado, o usuário deverá informar dentre as opções, a API de sua preferência, sendo ClimaTempo apenas nacionalmente, enquanto OpenWeather serve tanto para localidades nacionais quanto internacionais. (Figura. 11)



Informado o estado em letras maiúsculas, o bot solicitará a cidade (em letras minúsculas), que deverá ser informada pelo usuário. (Figura. 12)



Depois de o usuário informar todos os dados solicitados pelo AccuWeatherBot, ele informará o tempo atual do local informado, com data e horário, temperatura, sensação térmica, condição do céu, pressão, velocidade do vento, direção do vento e umidade relativa. (Figura. 13)

5 DIFICULDADES ENCONTRADAS DURANTE O TRABALHO

- Aprender do zero como cada ferramenta funciona e como utilizá-las dentro do código.
- Organização quanto as ideias que surgiam e como transformá-las na linguagem de programação.
- Dificuldade master = achar as restrições certas para que o programa funcionasse por passos.

6 SUGESTÕES DE TRABALHOS

- Chat bots, para ajudar usuários com sua praticidade, tendo infinitos usos e aplicações.
- Projetos de irrigação aliados a hardwares como Arduino ou RaspBerryPi.
- Identificação de fitopatologias através de fotos.
- Sistemas de gerenciamento de negócios.