**Capturando dados históricos climáticos com Python e API (OpenWeather)**

import requests

import mysql.connector

from datetime import datetime, timedelta

# Dados de acesso ao banco de dados

user = 'root'

password = '123456'

host = 'localhost'

database = 'dw\_salao\_de\_beleza'

# Chave de API OpenWeather

API\_KEY = "221f164d35f7154a527c7b3146fa2129"

# Coordenadas de Salvador

latitude = -12.9704

longitude = -38.5124

# Definindo a data inicial e a data final

data\_inicial = datetime(2023, 5, 2)

data\_final = datetime(2023, 5, 31)

# Função para converter de Kelvin para Celsius

def kelvin\_para\_celsius(temp\_kelvin):

    return temp\_kelvin - 273.15

# Função para obter dados climáticos da API OpenWeather

def obter\_temperatura(data):

    print("Iniciando coleta")

    link = f"https://api.openweathermap.org/data/3.0/onecall/day\_summary?lat={latitude}&lon={longitude}&date={data.strftime('%Y-%m-%d')}&appid={API\_KEY}"

    requisicao = requests.get(link)

    requisicao\_dic = requisicao.json()

    temperatura\_min = kelvin\_para\_celsius(requisicao\_dic['temperature']['min'])

    temperatura\_max = kelvin\_para\_celsius(requisicao\_dic['temperature']['max'])

    temperatura\_tarde = kelvin\_para\_celsius(requisicao\_dic['temperature']['afternoon'])

    temperatura\_noite = kelvin\_para\_celsius(requisicao\_dic['temperature']['night'])

    temperatura\_noite = kelvin\_para\_celsius(requisicao\_dic['temperature']['evening'])

    temperatura\_manha = kelvin\_para\_celsius(requisicao\_dic['temperature']['morning'])

    return temperatura\_min, temperatura\_max, temperatura\_tarde, temperatura\_noite, temperatura\_manha

# Loop para coletar os dados para cada data

while data\_inicial <= data\_final:

    temperatura\_min, temperatura\_max, temperatura\_tarde, temperatura\_noite, temperatura\_manha = obter\_temperatura(data\_inicial)

    if temperatura\_min is not None and temperatura\_max is not None and temperatura\_tarde is not None and temperatura\_noite is not None and temperatura\_manha is not None:

        # Conectar ao banco de dados

        db\_connection = mysql.connector.connect(

            host=host,

            user=user,

            password=password,

            database=database

        )

        cursor = db\_connection.cursor()

        # Inserir os dados no banco de dados

        insert\_query = "INSERT INTO d\_clima (temperatura\_min, temperatura\_max, temperatura\_tarde, temperatura\_noite, temperatura\_manha, dia, data\_atual) VALUES (%s, %s, %s, %s, %s, %s, %s)"

        data\_insercao = (temperatura\_min, temperatura\_max, temperatura\_tarde, temperatura\_noite, temperatura\_manha, data\_inicial.strftime('%Y-%m-%d'), datetime.now().strftime('%Y-%m-%d'))

        cursor.execute(insert\_query, data\_insercao)

        db\_connection.commit()

        # Fechar a conexão com o banco de dados

        cursor.close()

        db\_connection.close()

    # Incrementar a data inicial em um dia

    data\_inicial += timedelta(days=1)

print("Dados coletados e inseridos no banco de dados com sucesso.")

**Resumo:**

A captura dos dados climáticas se inicia com a inserção da chave API que se dá através do cadastro no site, <https://openweathermap.org/>, após isso foi necessário fazer uma requisição HTTPS que gera um JSON como resultado da coleta dos dados climáticos. Coletados os dados o código faz uma conexão com o Banco de Dados MySQL, quando conectado ele transforma os dados JSON chaves em colunas para inserção dos valores armazenados. Assim ele retorna um “Coletando dados” quando inicia a coleta e “Dados coletados e inseridos no banco de dados com sucesso” quando a inserção no banco não retorna erro.

**Tabela dimensão D\_clima:**

*\*\*A tabela d\_clima iniciou sem dados*

**Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente**

**Executando:**

**Tela de computador com fundo preto

Descrição gerada automaticamente**

**Tela de computador com fundo preto

Descrição gerada automaticamente**

**Resultado:**

Consideramos satisfatórios os resultado da coleta, quando foi retornado o resultado JSON

* Temperatura mínima
* Temperatura máxima
* Temperatura de noite
* Temperatura de manhã
* Dia que foi extraído os dados históricos
* Data atual da requisição

Obs: Será implementado no código a capacidade de verificar dados coletados, e atribuir uma nova API para obter dados da previsão do dia que foi feita a requisição HTTP.

**Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente**

**API OpenWeather:**

OpenWeather é uma equipe de especialistas em TI e cientistas de dados que pratica ciência profunda de dados meteorológicos. Para cada ponto do globo, o OpenWeather fornece dados meteorológicos históricos, atuais e previstos por meio de APIs de velocidade da luz. Sede em Londres, Reino Unido.

**IDE:**

Visual Studio Code

**Linguagem de Programação:**

Python