

## 01) Extraindo as informações da API

```
import requests

API_KEY = "XXXXXXXX"

url = f"https://api.hgbrasil.com/finance?key={API_KEY}"

try:
    response = requests.get(url)
    response.raise_for_status()

    data = response.json()

    dolar = data['results']['currencies']['USD']
    euro = data['results']['currencies']['EUR']

    print("=== COTAÇÕES ATUAIS ===")
    print(f"Dólar: R$ {dolar['buy']:.2f}")
    print(f"Euro: R$ {euro['buy']:.2f}")
    print(f"Última atualização: {data['results']['currencies']['source']}")

except requests.exceptions.RequestException as e:
    print("Erro ao acessar a API:", e)
except KeyError:
    print("Erro ao processar os dados recebidos. Verifique a estrutura da resposta.")
```

## 02) Criando a tabela para armazenar as informações

```
def salvar_cotacoes(dolar, euro):

    conexao = sqlite3.connect("bdcotacoes.db")
    cursor = conexao.cursor()

    cursor.execute("""
        CREATE TABLE IF NOT EXISTS moedas (
            Data TEXT,
            Dolar REAL,
            Euro REAL
        )
    """)

    data_atual = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    cursor.execute("INSERT INTO moedas (Data, Dolar, Euro) VALUES (?, ?, ?)",
        (data_atual, dolar, euro))

    conexao.commit()
    conexao.close()
```

03) Alterando o arquivo para realizar as mudanças sempre que executar o arquivo

```
import requests
import sqlite3
from datetime import datetime

API_KEY = "XXXXXXXX"
URL = f"https://api.hgbrasil.com/finance?format=json-cors&key={API_KEY}"

def buscar_cotacoes():
    try:
        r = requests.get(URL)
        dados = r.json()

        dolar = dados['results']['currencies']['USD']['buy']
        euro = dados['results']['currencies']['EUR']['buy']

        return dolar, euro
    except Exception as e:
        print("Erro ao buscar cotações:", e)
        return None, None

def salvar_cotacoes(dolar, euro):

    conexao = sqlite3.connect("bdcotacoes.db")
    cursor = conexao.cursor()

    cursor.execute("""
        CREATE TABLE IF NOT EXISTS moedas (
            Data TEXT,
            Dolar REAL,
            Euro REAL
        )
    """)

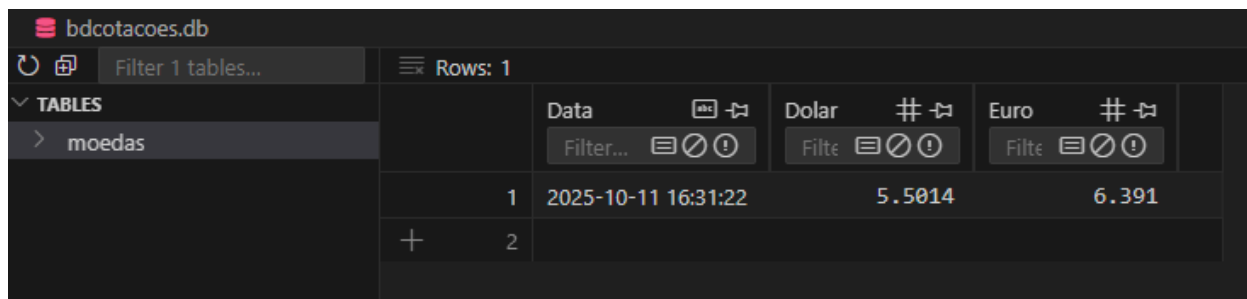
    data_atual = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    cursor.execute("INSERT INTO moedas (Data, Dolar, Euro) VALUES (?, ?, ?)",
                   (data_atual, dolar, euro))

    conexao.commit()
    conexao.close()
```

```
def main():
    dolar, euro = buscar_cotacoes()
    if dolar is not None and euro is not None:
        salvar_cotacoes(dolar, euro)
        print("Cotações salvas com sucesso!")
        print(f"Dólar: R$ {dolar:.2f} | Euro: R$ {euro:.2f}")
    else:
        print("Não foi possível salvar as cotações.")

if __name__ == "__main__":
    main()
```

#### 04) Resultado do banco de dados



The screenshot shows a database application window titled 'bdcotacoes.db'. The interface includes a sidebar with a 'TABLES' section containing a table named 'moedas'. The main area displays the table's data. The table has three columns: 'Data', 'Dolar', and 'Euro'. The first row shows the date '2025-10-11 16:31:22', a Dolar rate of '5.5014', and an Euro rate of '6.391'. The second row is partially visible, starting with a '+' icon and the number '2'.

|     | Data                | Dolar  | Euro  |
|-----|---------------------|--------|-------|
| 1   | 2025-10-11 16:31:22 | 5.5014 | 6.391 |
| + 2 |                     |        |       |