



BANCO DE DADOS AVANÇADOS
PROJETO BOLSA FAMÍLIA
MARÇO/23

ALUNOS: ANA TEREZA, JOÃO LOPES, GUILHERME HENRIQUE

RA: 60683, 63439, 63717

FERRAMENTAS UTILIZADAS

Linguagem de programação:
Python

Bibliotecas nativas
OS
TIME
ZIPFILE

Bibliotecas externas:

Pandas
Psycopg2
Sqlalchemy
Psutil

SGBD
POSTGRE versão 14

ChatGPT para correção de erros

Biblioteca **OS** em Python

A biblioteca **os** é uma ferramenta poderosa para automação e interação com o sistema operacional em Python, permitindo que você manipule arquivos, diretórios e execute comandos de forma simples e eficiente. Ela é amplamente usada em scripts de automação e tarefas que requerem integração direta com o sistema.

PRINCIPAIS FUNCIONALIDADES

Manipulação de Arquivos e Diretórios:

- `os.open()`, `os.close()`: Gerenciam a abertura e o fechamento de arquivos em baixo nível.
- `os.getcwd()`: Retorna o diretório atual.
- `os.listdir()`, `os.mkdir()`, `os.makedirs()`: Listam e criam arquivos e diretórios.
- `os.remove()`, `os.rmdir()`, `os.removedirs()`: Removem arquivos e diretórios.

Variáveis de Ambiente:

- `os.getenv()`: Obtém valores de variáveis de ambiente.
- `os.environ`: Acessa e define variáveis de ambiente, útil para configurações sensíveis.

Caminhos de Arquivos:

- `os.path.join()`: Combina caminhos, evitando erros entre diferentes sistemas operacionais.
- `os.path.exists()`, `os.path.isfile()`, `os.path.isdir()`: Verificam existência e tipo de arquivo ou diretório.

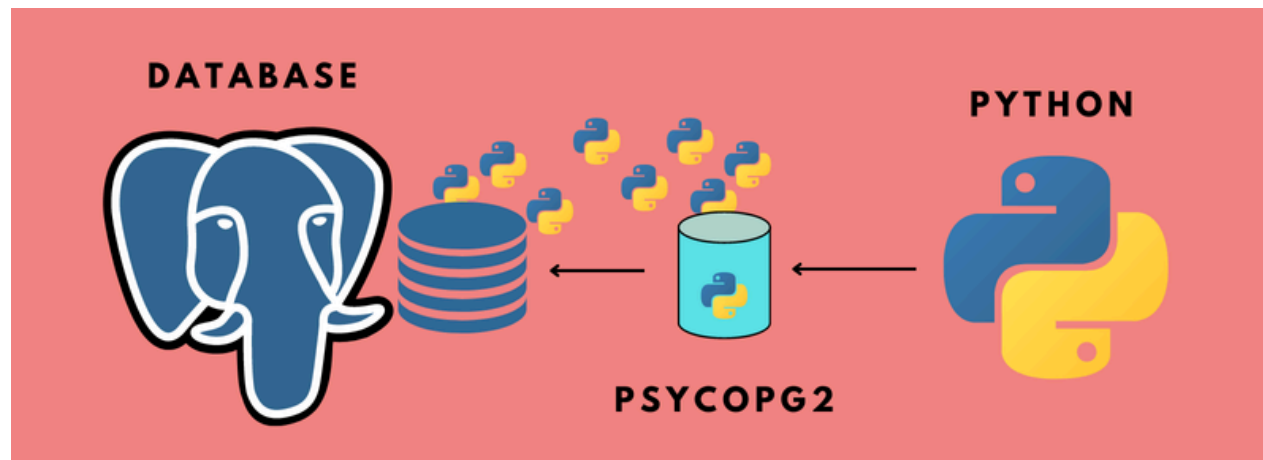
Execução de Comandos no Sistema:

- `os.system()`: Executa comandos diretamente no sistema.
- `os.exec*()` e `os.spawn*()`: Para processos mais complexos.

Psycopg2

O que é o Psycopg2?

O psycopg2 é uma biblioteca Python popular que permite a interação com bancos de dados PostgreSQL. Ele implementa a especificação da API DB para Python, facilitando a execução de consultas SQL, a recuperação de dados e o gerenciamento de transações no PostgreSQL diretamente a partir de scripts ou aplicativos Python.

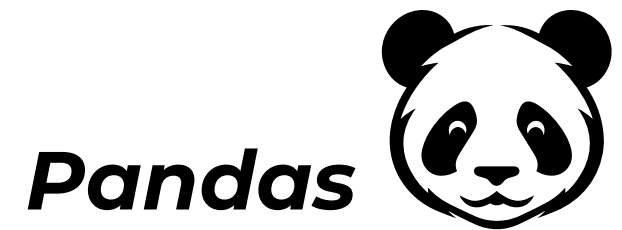


SQLAlchemy

O SQLAlchemy é uma biblioteca Python poderosa para trabalhar com bancos de dados relacionais, como PostgreSQL, MySQL, SQLite e outros. Ela fornece uma interface de alto nível que simplifica a manipulação de bancos de dados, abstraindo a interação direta com SQL e oferecendo duas abordagens principais: Core SQL Expression Language e Object-Relational Mapper

Psutil

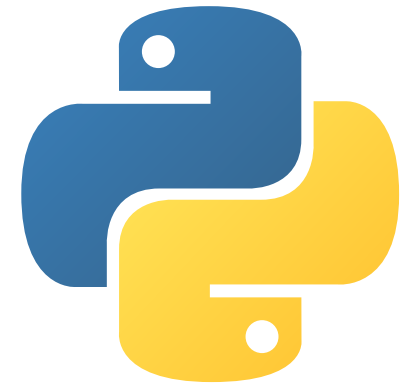
É um módulo Python que monitora o sistema operacional e os processos, permitindo recuperar informações sobre a utilização do sistema e a execução de processos. Pode monitorar: Consumo de CPU, Consumo de memória, Consumo de rede, Discos, Processos e seu consumo individual de CPU e memória.



A biblioteca pandas é uma ferramenta poderosa para manipulação e análise de dados em Python. Com ela, você pode trabalhar com grandes volumes de dados de maneira eficiente e intuitiva. O pandas é amplamente utilizado em ciência de dados, análise de dados e aprendizado de máquina devido à sua capacidade de organizar, limpar e manipular dados tabulares.

ZIPFILE

É uma biblioteca padrão que permite trabalhar com arquivos ZIP, como compactar e descompactar arquivos.



TIME

É uma biblioteca do Python que te permite trabalhar com datas e horas

Principais problemas que enfrentamos

- *Dados não importados*
- *Erros não apareciam nos logs do banco de dados*
- *Erro ao importar grandes quantidades de dados do arquivo cvs, alguns arquivos não estão bem formatados para as tabelas criadas no banco de dados, exemplo: dados to topo float no postgre são separados por . E as colunas das tabelas por , e nos ponto e vírgula.*
- *Alta utilização da memória RAM do computador*
- *Uso excessivo da CPU*
- *Armazenamento inadequado do arquivo csv*
- *Erro de compilação, BIOS do programador, fazendo cagada, errando variáveis, objetos ou blocos de código*




CONFIGURAÇÕES DA MÁQUINA

Sistema > **Sobre**

LAPTOP-5R0D7027
550XDA

Renomear este computador

 Especificações do dispositivo

Copiar ^


| | |
|---------------------|--|
| Nome do dispositivo | LAPTOP-5R0D7027 |
| Processador | Intel(R) Celeron(R) 6305 @ 1.80GHz 1.80 GHz |
| RAM instalada | 4,00 GB (utilizável: 3,73 GB) |
| ID do dispositivo | 47D57F4A-DFD2-4C89-A849-28579721BD63 |
| ID do Produto | 00327-60000-00000-AA091 |
| Tipo de sistema | Sistema operacional de 64 bits, processador baseado em x64 |
| Caneta e toque | Nenhuma entrada à caneta ou por toque disponível para este vídeo |

Links relacionados

[Domínio ou grupo de trabalho](#)

[Proteção do sistema](#)

[Configurações avançadas do sistema](#)

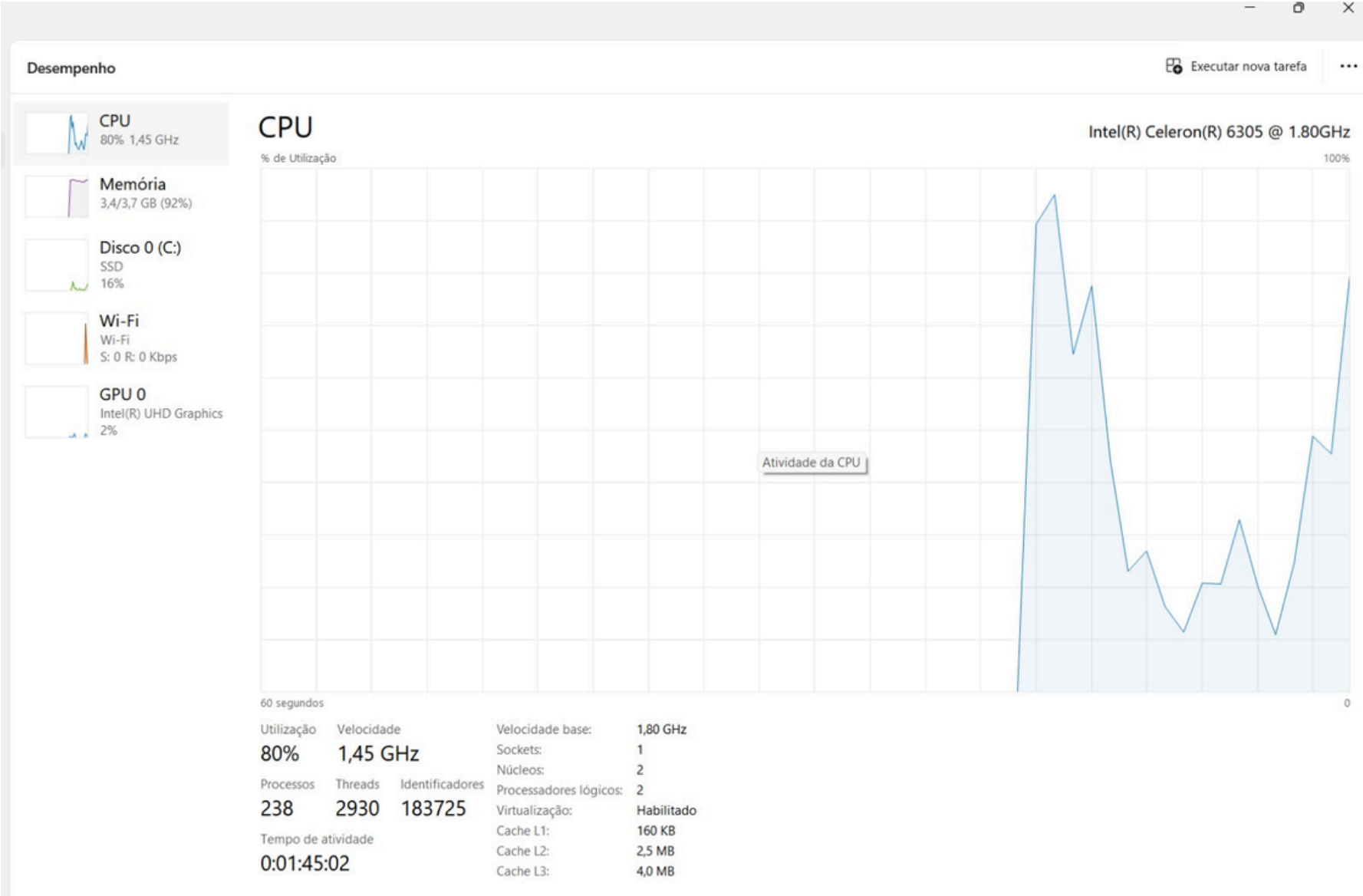
 Especificações do Windows

Copiar ^

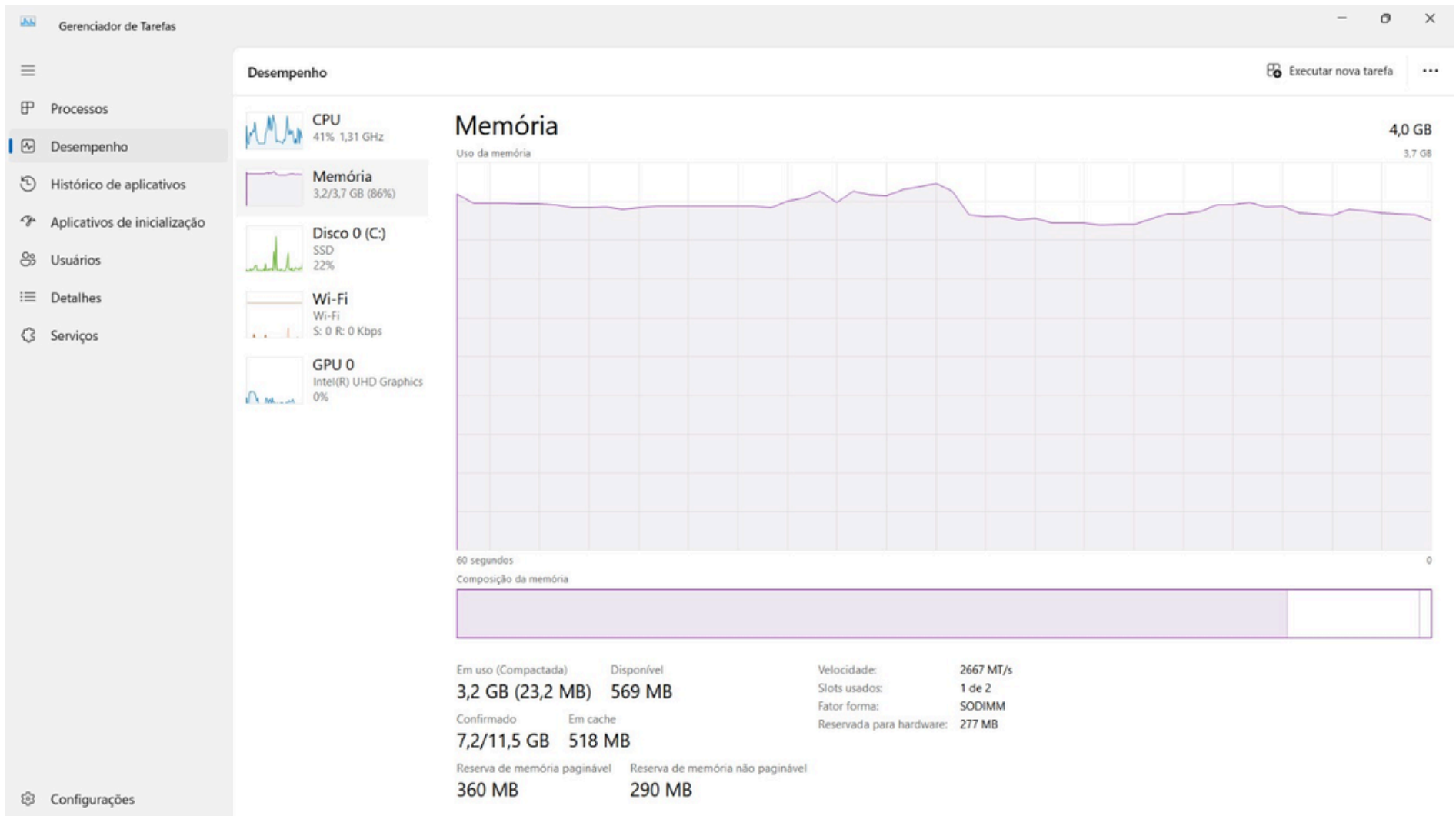
| | |
|------------------|---|
| Edição | Windows 11 Home Single Language |
| Versão | 23H2 |
| Instalado em | 07/03/2023 |
| Compilação do SO | 22631.4391 |
| Experiência | Windows Feature Experience Pack 1000.22700.1047.0 |

[Contrato de Serviços Microsoft](#)
[Termos de Licença de Software da Microsoft](#)

ESPECIFICAÇÕES DA CPU



ESPECIFICAÇÕES DA MEMÓRIA RAM



DESCOMPACTAR ZIP

Listando o arquivo.csv:

202403_NovoBolsaFamilia.csv

Arquivo extraído para o caminho:

C:\Users\FABIA\Downloads\202403_NovoBolsaFamilia.csv

Tempo de extração: 31.64 segundos

Conexão realizada

INSERÇÃO DOS DADOS

34,77 minutos para inserir todos os ***20.492.878***
de dados no banco, já devidamente tratados.

BIBLIOTECAS IMPORTADAS

import zipfile

import pandas as pd

from sqlalchemy

import create_engine

import time

import os

import psutil

TABELA DO BANCO DE DADOS

```
CREATE TABLE pagamentos(  
    data_referencia VARCHAR(100),  
    data_competencia VARCHAR(100),  
    uf VARCHAR(5),  
    codigo_municipio VARCHAR(100),  
    nome_municipio VARCHAR(100),  
    cpf_beneficiario VARCHAR(100),  
    nis_beneficiario VARCHAR(100),  
    nome_beneficiario VARCHAR(100),  
    valor_parcela DECIMAL (10,2)  
);
```

CONSULTAS

Total de Registros

```
SELECT COUNT(*)
FROM pagamentos;
```

```
Query    Query History
32
33  /* Seleccionando todos os registros da tabela pagamentos*/
34  SELECT COUNT(*) FROM pagamentos;
35
```

Data Output Messages Notifications

Icons: Expand, Copy, Paste, Undo, Redo, Delete, Refresh, Download, SQL

| | count bigint | |
|---|-----------------|-------------|
| 1 | 20492879 | 46 47 /* |

Tempo 9.114s

Soma Total de Valores Pagos

```
SELECT SUM(valor_parcela)
FROM pagamentos;
```

```
46
47  /*Selecionando o total de valores pagos no mês referente de março*/
48  SELECT SUM(valor_parcela) FROM pagamentos;
```

Data Output Messages Notifications

Icons:

| | sum numeric |
|---|----------------|
| 1 | 13373651126.00 |

Tempo 22.067s

Total de beneficiados para cidade de Brasília, São Paulo e Salvador

```
65
66 -- Mesma coisa do de cima kkk, só que todo mundo junto
67 v SELECT nome_municipio, COUNT(nome_beneficiario) FROM pagamentos
68     WHERE nome_municipio IN('SAO PAULO','BRASILIA','SALVADOR')
69     GROUP BY nome_municipio;
70
71 -- Contabilizar total de parcelas pagas as cidades BRASILIA, SAO PAULO, SALVADOR
72 v SELECT nome_municipio, SUM(valor_parcela) FROM pagamentos
```

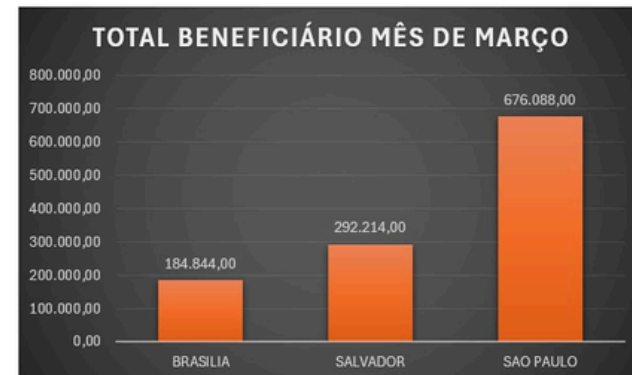
Data Output Messages Notifications



| | nome_municipio character varying (100) | count bigint |
|---|---|-----------------|
| 1 | BRASILIA | 184844 |
| 2 | SALVADOR | 292214 |
| 3 | SAO PAULO | 676088 |

Tempo 21.739s

| MUNICIPIO | TOTAL BENEFICIÁRIO |
|-----------|--------------------|
| BRASILIA | 184.844,00 |
| SALVADOR | 292.214,00 |
| SAO PAULO | 676.088,00 |



Soma total dos valores pagos por município ordenado por ordem alfabética.

```
75
76  /*Retorna a soma total de parcela por município, ordenado em ordem alfabetica */
77  SELECT nome_municipio, uf, SUM(valor_parcela) AS total_valor_pago
78  FROM pagamentos
79  GROUP BY nome_municipio, uf
80  ORDER BY nome_municipio;
81  -- Esse cálculo retornou em 32.99s
82
83  /* Retorna os municipios que mais receberam */
```

Data Output Messages Notifications

SQL

| | nome_municipio character varying (100) | uf character varying (5) | total_valor_pago numeric |
|---|---|-----------------------------|-----------------------------|
| 1 | ABADIA DE GOIAS | GO | 1323881.00 |
| 2 | ABADIA DOS DOURADOS | MG | 267347.00 |
| 3 | ABADIANIA | GO | 976513.00 |
| 4 | ABAETE | MG | 1034411.00 |
| 5 | ABAETETUBA | PA | 26618580.00 |
| 6 | ABAIARA | CE | 1355583.00 |
| 7 | ABAIRA | BA | 1236966.00 |
| 8 | ABARE | BA | 3056435.00 |

Total rows: 1000 of 5570 Query complete 00:00:26.912 Ln 77, Col 1

Tempo 26.912s

Soma total dos valores pagos por município ordenado por maiores valores recebidos.

```
82
83  /* Retorna os municipios que mais receberam */
84  ✓ SELECT nome_municipio, uf, SUM(valor_parcela) AS total_valor_pago
85  FROM pagamentos
86  GROUP BY nome_municipio, uf
87  ORDER BY total_valor_pago DESC;
88  -- esse calculo retornou em 28.44s
89
90  /* Retorna uma lista de cpf repetidos */
91  ✓ SELECT cpf_beneficiario, COUNT(*) AS quantidade
92  FROM pagamentos
93  GROUP BY cpf_beneficiario
```

Data Output Messages Notifications

SQL

| | nome_municipio character varying (100) | uf character varying (5) | total_valor_pago numeric |
|---|---|-----------------------------|-----------------------------|
| 1 | SAO PAULO | SP | 435806841.00 |
| 2 | RIO DE JANEIRO | RJ | 350814041.00 |
| 3 | FORTALEZA | CE | 210948395.00 |
| 4 | SALVADOR | BA | 182642906.00 |
| 5 | MANAUS | AM | 178603995.00 |
| 6 | BRASILIA | DF | 121619766.00 |
| 7 | BELEM | PA | 114933731.00 |
| 8 | RECIFE | PE | 90836423.00 |

Total rows: 1000 of 5570 Query complete 00:00:31.436 Ln 83, Col 1

Tempo 31.436s

Identificar se existem CPF repetidos no banco de dados

```
89
90 /* Retorna uma lista de cpf repetidos */
91 v SELECT cpf_beneficiario, COUNT(*) AS quantidade
92 FROM pagamentos
93 GROUP BY cpf_beneficiario
94 HAVING COUNT(*) > 1
95 ORDER BY quantidade DESC;
96 -- Retornou resultado em 1 min e 51s, retornou 3.515.664 de valores null
97
98 -- Retornando o valor total de parcelas pagas por estado
99 v SELECT DISTINCT uf, SUM(valor parcela) FROM pagamentos
```

Data Output Messages Notifications

| | cpf_beneficiario character varying (100) | quantidade bigint |
|---|---|----------------------|
| 1 | [null] | 3515664 |
| 2 | ***.088.303-** | 59 |
| 3 | ***.222.103-** | 57 |
| 4 | ***.087.103-** | 54 |
| 5 | ***.596.503-** | 53 |
| 6 | ***.855.834-** | 53 |
| 7 | ***.184.634-** | 52 |
| 8 | ***.170.324-** | 52 |

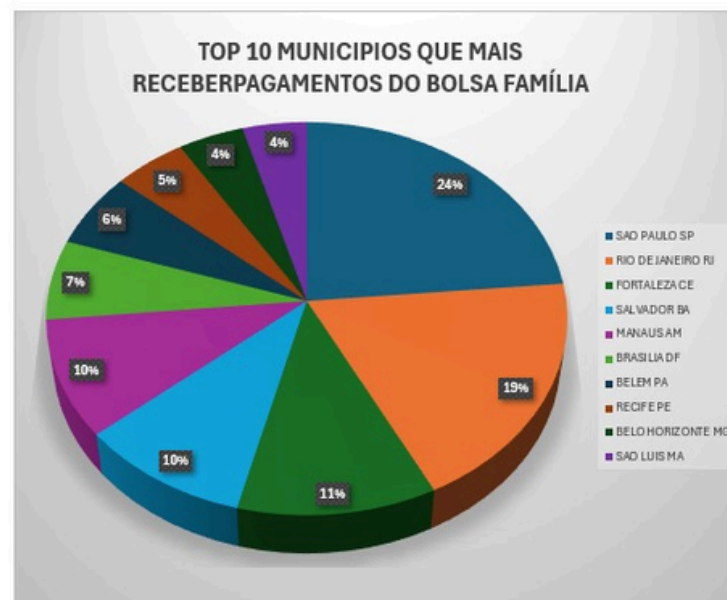
Total rows: 1000 of 993372 Query complete 00:01:52.149 Ln 91, Col 1

Tempo 1min e 51seg

* vários CPF nulos e repetidos

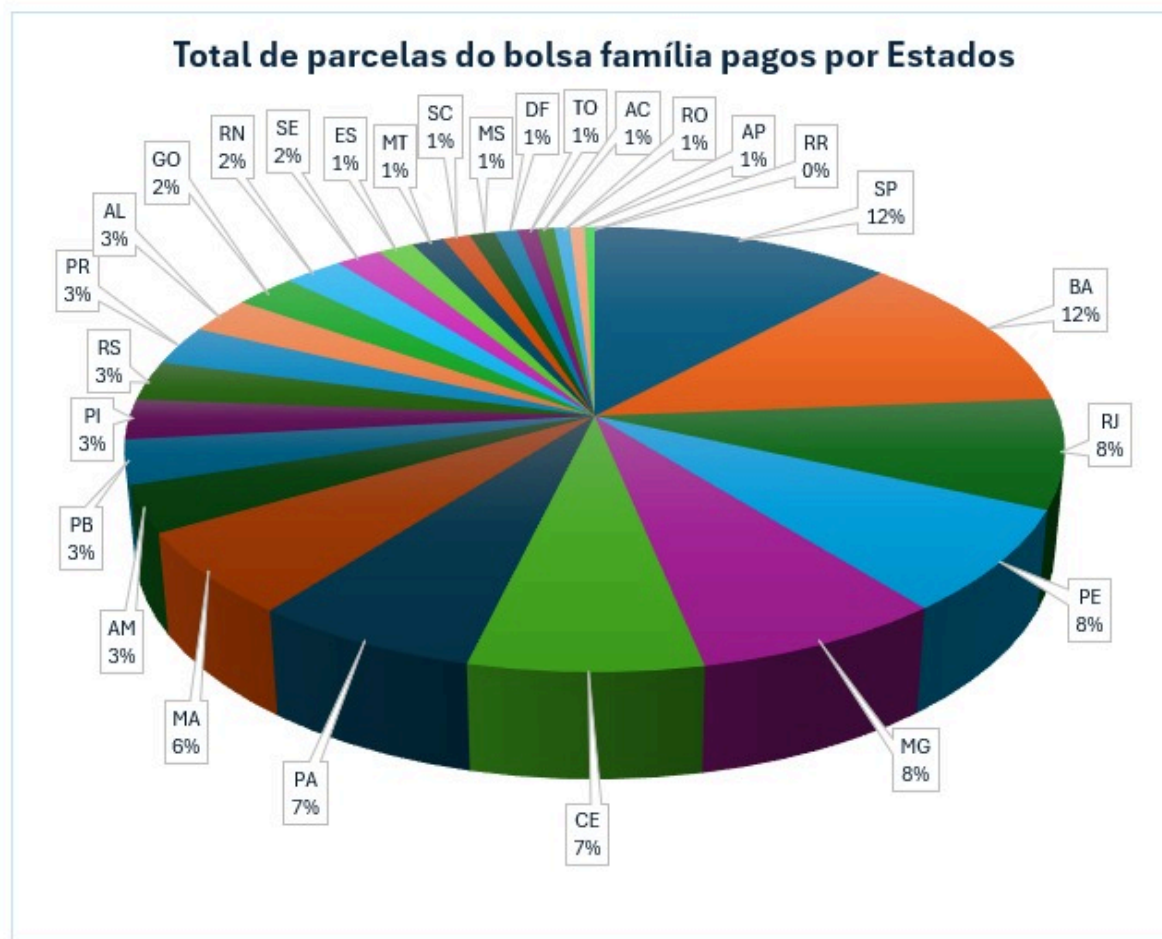
10 MUNICÍPIOS QUE MAIS RECEBERAM PAGAMENTOS DO BOLSA FAMÍLIA

| MUNICÍPIO | UF | TOTAL DE PAGAMENTOS REALIZADO |
|-------------------------|----|-------------------------------|
| SAO PAULO | SP | R\$ 435.806.841,00 |
| RIO DE JANEIRO | RJ | R\$ 350.814.041,00 |
| FORTALEZA | CE | R\$ 210.948.395,00 |
| SALVADOR | BA | R\$ 182.642.906,00 |
| MANAUS | AM | R\$ 178.603.995,00 |
| BRASILIA | DF | R\$ 121.619.766,00 |
| BELEM | PA | R\$ 114.933.731,00 |
| RECIFE | PE | R\$ 90.836.423,00 |
| BELO HORIZONTE | MG | R\$ 82.462.540,00 |
| SAO LUIS | MA | R\$ 80.377.379,00 |
| NOVA IGUACU | RJ | R\$ 78.560.363,00 |
| DUQUE DE CAXIAS | RJ | R\$ 76.008.528,00 |
| MACEIO | AL | R\$ 71.059.424,00 |
| GUARULHOS | SP | R\$ 68.734.353,00 |
| TERESINA | PI | R\$ 65.108.608,00 |
| JABOATAO DOS GUARARAPES | PE | R\$ 60.772.787,00 |
| JOAO PESSOA | PB | R\$ 53.641.687,00 |
| SAO GONCALO | RJ | R\$ 51.795.259,00 |
| PORTO ALEGRE | RS | R\$ 50.257.448,00 |
| NATAL | RN | R\$ 50.203.393,00 |
| GOIANIA | GO | R\$ 48.536.644,00 |
| FEIRA DE SANTANA | BA | R\$ 47.706.727,00 |
| BELFORD ROXO | RJ | R\$ 47.234.262,00 |
| CAUCAIA | CE | R\$ 41.514.317,00 |
| CAMPOS DOS GOYTACAZES | RJ | R\$ 38.924.365,00 |
| CAMPINAS | SP | R\$ 38.052.506,00 |
| ARACAJU | SE | R\$ 37.627.424,00 |
| MACAPA | AP | R\$ 37.340.563,00 |
| CURITIBA | PR | R\$ 37.337.280,00 |
| CAMPO GRANDE | MS | R\$ 36.284.139,00 |
| OLINDA | PE | R\$ 35.405.742,00 |
| SANTAREM | PA | R\$ 34.305.339,00 |
| CARUARU | PE | R\$ 32.077.621,00 |
| VITORIA DA CONQUISTA | BA | R\$ 31.965.045,00 |
| PETROLINA | PE | R\$ 30.816.980,00 |
| ANANINDEUA | PA | R\$ 30.362.779,00 |
| OSASCO | SP | R\$ 29.524.895,00 |
| PAULISTA | PE | R\$ 29.450.739,00 |
| CAMACARI | BA | R\$ 28.595.901,00 |
| RIO BRANCO | AC | R\$ 28.326.534,00 |
| CONTAGEM | MG | R\$ 28.143.753,00 |
| PORTO VELHO | RO | R\$ 27.307.677,00 |
| CUIABA | MT | R\$ 27.297.255,00 |
| ABAETETUBA | PA | R\$ 26.618.580,00 |
| SAO JOAO DE MERITI | RJ | R\$ 26.594.774,00 |
| SAO BERNARDO DO CAMPO | SP | R\$ 26.463.198,00 |
| SERRA | ES | R\$ 26.433.241,00 |



TOTAL DE PARCELAS DO BOLSA FAMÍLIA PAGOS POR ESTADO

| UF | VALOR_TOTAL_PARCELA |
|-------|-----------------------|
| SP | R\$ 1.602.517.308,00 |
| BA | R\$ 1.570.083.095,00 |
| RJ | R\$ 1.061.835.995,00 |
| PE | R\$ 1.012.956.262,00 |
| MG | R\$ 1.006.484.529,00 |
| CE | R\$ 936.884.282,00 |
| PA | R\$ 906.362.204,00 |
| MA | R\$ 821.414.512,00 |
| AM | R\$ 454.796.150,00 |
| PB | R\$ 429.668.392,00 |
| PI | R\$ 389.966.025,00 |
| RS | R\$ 385.865.306,00 |
| PR | R\$ 382.454.282,00 |
| AL | R\$ 351.844.159,00 |
| GO | R\$ 329.236.720,00 |
| RN | R\$ 314.516.777,00 |
| SE | R\$ 241.976.414,00 |
| ES | R\$ 192.806.351,00 |
| MT | R\$ 170.475.072,00 |
| SC | R\$ 144.189.268,00 |
| MS | R\$ 135.567.008,00 |
| DF | R\$ 121.619.766,00 |
| TO | R\$ 105.159.407,00 |
| AC | R\$ 90.596.769,00 |
| RO | R\$ 81.421.345,00 |
| AP | R\$ 80.078.590,00 |
| RR | R\$ 52.875.138,00 |
| TOTAL | R\$ 13.373.651.126,00 |



TOTAL DE PAGAMENTOS REALIZADOS

| CIDADE | TOTAL DE PAGAMENTOS REALIZADOS |
|-----------|--------------------------------|
| BRASILIA | 121.619.766,00 |
| SALVADOR | 182.642.906,00 |
| SAO PAULO | 435.806.841,00 |



OBRIGADA PELA ATENÇÃO!

AQUI TEMOS O VERDADEIRO

