

# Projetos candidatos política do Estado de Goiás

Documentação do projeto.

Autora: Nínive

Local de extração dos dados:



Link: <https://dadosabertos.tse.jus.br/dataset/?groups=candidatos>

Os dados extraídos são de 1994 até 2022 do estado de Goiás, os dados são sobre os candidatos das eleições pelo estado do Goiás.

Dicionário dos dados Link:

[file:///C:/Users/niniv/Downloads/consulta\\_cand\\_1994/leiamme.pdf](file:///C:/Users/niniv/Downloads/consulta_cand_1994/leiamme.pdf)

## Etapas realizadas no projeto

1. Extração dos csv's na página citada acima. Limpeza dos dados após uma análise exploratória dos dados.
2. Junção dos csv's do estado de Goiás referente aos anos de 1994 até 2022, cada csv foi aberto e limpo, ao final foi feito uma junção para ficar um csv único.
3. Criação de um banco de dados pra armazenar esses dados no csv.
4. Criação de uma API que faz as consultas dos dados armazenados no banco.
5. Criação de um painel para a visualização desses dados.

## Detalhes de cada etapa

Etapa 1:

Na etapa 1 foi feito a extração dos dados, após isso foi utilizado a ide Jupyter e a linguagem Python e a biblioteca Pandas.

Toda a etapa de abertura dos csv's e limpeza estão detalhados nos arquivos que estão na pasta do repositório onde estão o projeto. No link:

[https://github.com/ninivehelen/projeto\\_candidatos\\_politica](https://github.com/ninivehelen/projeto_candidatos_politica) . A pasta candidatos\_goiás\_1994ate\_2008 e a pasta candidatos\_goiás\_2010ate\_2022 contém um arquivo chamado candidatos\_goiás\_1994\_2008.ipynb e candidatos\_goiás\_2010\_2022.ipynb que contém toda a exploração dos dados e as limpezas realizadas.

Etapa 2:

Após a limpeza foi realizada a junção dos csv's extraídos, para ficar um arquivo único para inserir os dados no banco de dados. Utilizando a linguagem python e o visual studio para a criação do código para juntar os arquivos. Nesta etapa esta sendo usado os arquivos que

estão na pasta chamada `scripts_py` . Nesta pasta contém o código que fez a junção dos arquivos e os demais códigos das outras etapas.

#### Etapa 3:

Após a junção dos dados. Foi utilizando ainda a linguagem python para conectar ao mysql, que é um banco de dados, e a criação de um banco chamado `candidatos_goiás`. Depois do banco ser criado o csv único citado nas etapas acima, foi utilizado para ler os dados do csv dele e inserir no banco de dados.

#### Etapa 4:

Depois dos dados inseridos ao banco de dados, foi criada uma API para consultar os dados inserido no banco. Para rodar a API, é necessário ter rodado toda as etapas acima. Rodando a API construída é possível fazer consulta nos dados através uma url.

## Como rodar o código

### Instruções pra rodar o código

1. Para rodar o código você precisa ter o mysql instalado na sua máquina, junto com o Python e uma IDE de sua preferência.
2. Após ter tudo instalado citado acima, você pode baixar o projeto no repositório neste link: [https://github.com/ninivehelen/projeto\\_candidatos\\_politica](https://github.com/ninivehelen/projeto_candidatos_politica), se você tem o git você pode também fazer o clone do projeto. Fica a seu critério.
3. Após ter baixado o código, abra ele em sua IDE de sua preferência e que rode python.
4. Após abrir o projeto em sua IDE, você pode rodar o arquivo `Requests.txt`, ele vai instalar todas as dependências necessárias para rodar o projeto.
5. Agora, está na hora de configurar as credencias do seu banco de dados, pra isso eu deixei um arquivo dentro do projeto na pasta `scripts.py` chamado `credencia_banco.yaml`, com ele você fará as credencias do seu banco, ao abrir ele na sua IDE, coloque o nome do seu host, se seu host for localhost então não precisa alterar, coloque o nome do seu user, se o seu user já for root, então não vai precisar alterar, agora coloque a sua senha do seu banco, se seu banco não possui senha, pode deixar em branco, agora em database não mude nada, pois esse banco `candidatos_goiás` será criado automaticamente. Agora que já configurou as credencias tudo ok é hora de rodar o código
6. Rode o arquivo `main.py`, esse arquivo fara a leitura dos csv já limpos e fará a união deles, depois disso esse código vai fazer a chamada do outro arquivo chamado `criar_banco.py` , que fará a criação do banco de dados chamada. `Candidatos_goiás`, após a criação do banco, ira se chamada automaticamente também outro arquivo chamado, `salvar_dados.py`, esse arquivo vai ler o csv final unido e abrir ele e ler linha a linha e salvar no banco de dados, neste momento vai aparecer dados inserido varias vezes, aguarde ate o final da execução, essa etapa leva bastante tempo.
7. Após os dados serem inseridos, agora é hora de rodar a API para consultar os dados inseridos. Para rodar a API é necessários que todas as outras etapas tenham sido compiladas de maneira correta. Dentro da pasta `script_py` tem um arquivo chamado `API_candidatos.py`, rode esse arquivo. Após rodar o arquivo perceba que aparece um endereço chamado: <http://127.0.0.1:5000> se esse endereço apareceu, é por que o código rodou adequadamente.

## Utilizando a API

Utilizando a api dos dados de candidatos do estado de Goiás

Para utilizar a API para fazer consultas é necessário que o código do arquivo citado na etapa acima, na parte 7 esteja rodando. Com o link chamado <http://127.0.0.1:5000> que aparece ao rodar o código API\_candidatos.py. Segue exemplos para utilizar

No seu navegador ou utilizando o POSTMAN que serve para testar API, faça o exemplo a seguir

### Primeira Consulta

Se estiver no seu navegador cole o seguinte e aperte Enter.

[http://127.0.0.1:5000/candidatos\\_goias](http://127.0.0.1:5000/candidatos_goias)

Esse caminho vai trazer todos os dados de candidatos de todos os anos e todas as informações do estado de Goiás.

### Segunda consulta

Na primeira consulta foi mostrado os candidatos e todas as informações do estado de Goiás, nesta próxima consulta você poderá consultar agora por nome do candidato, na consulta acima, veja um nome de algum candidato e digite no lugar de nomecandidato dentro de aspas na url a seguir, você pode escolher um, mas aqui vai umas duas sugestão para pesquisar:

BENIVALDO RIBEIRO DE SOUZA

CLELIA BRANDAO ALVARENGA CRAVEIRO

[http://127.0.0.1:5000/candidatos\\_goias/eleicao/nome\\_candidato/nomecandidato](http://127.0.0.1:5000/candidatos_goias/eleicao/nome_candidato/nomecandidato)

Exemplo:

[http://127.0.0.1:5000/candidatos\\_goias/eleicao/nome\\_candidato/ CLELIA BRANDAO ALVARENGA CRAVEIRO](http://127.0.0.1:5000/candidatos_goias/eleicao/nome_candidato/CLELIA BRANDAO ALVARENGA CRAVEIRO)

OBS: coloque dentro de aspas quando for rodar no navegador o nome do candidato na url

### Terceira Consulta

Nesta próxima consulta você poderá pesquisar por ano da eleição, a pesquisa vai mostrar informações dos candidatos e das eleições no ano que você filtrar.

No seu navegador digite a url a seguir, e no lugar de anoeleicao digite o ano da eleição que deseja pesquisar. Pode ser entre 1994 até 2022

[http://127.0.0.1:5000/candidatos\\_goias/eleicao/ano\\_eleicao/anoeleicao](http://127.0.0.1:5000/candidatos_goias/eleicao/ano_eleicao/anoeleicao)

Exemplo:

[http://127.0.0.1:5000/candidatos\\_goias/eleicao/ano\\_eleicao/1996](http://127.0.0.1:5000/candidatos_goias/eleicao/ano_eleicao/1996)

### Quarta Consulta

Agora poderá pesquisar por situação da eleição: Situações existentes:

ELEITO, NÃO ELEITO, SUPLENTE, e outras situações que podem ser procuradas no dicionário dos dados para pesquisa.

Exemplo da url para pesquisar pela situação da eleição do candidato

Digite a url a seguir, e no local de situacao, digite uma situação, ELEITO, NÃO ELEITO, SUPLENTE, no lugar de situacao entre aspas

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/situacao\\_candidato/situacao](http://127.0.0.1:5000/candidatos_goiás/eleicao/situacao_candidato/situacao)

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/situacao\\_candidato/'SUPLENTE'](http://127.0.0.1:5000/candidatos_goiás/eleicao/situacao_candidato/'SUPLENTE')

rode no seu navegador a url, e você verá que aparecerá a consulta.

Agora que já ficou mais entendível. Tem esses outros exemplos de consulta que podem ser pesquisados via API

Escolaridade:

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/escolaridade\\_candidato/'ENSINO MÉDIO COMPLETO'](http://127.0.0.1:5000/candidatos_goiás/eleicao/escolaridade_candidato/'ENSINO MÉDIO COMPLETO')

Existe outras escolaridades que podem ser pesquisadas, é só substituir no final da url 'ENSINO MÉDIO COMPLETO' por uma das escolaridades a seguir:

ENSINO MEDIO COMPLETO

ENSINO MÉDIO INCOMPLETO

ENSINO FUNDAMENTAL COMPLETO

ENSINO FUNDAMENTAL INCOMPLETO

SUPERIOR COMPLETO

SUPERIOR INCOMPLETO

LÊ E ESCRIVE

Sigla:

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/sigla\\_partido\\_candidato/'sigladopartido'](http://127.0.0.1:5000/candidatos_goiás/eleicao/sigla_partido_candidato/'sigladopartido')

onde está escrito sigladopartido entre aspas, substitua por uma sigla de um partido existente.

Genero:

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/genero\\_candidato/'FEMININO'](http://127.0.0.1:5000/candidatos_goiás/eleicao/genero_candidato/'FEMININO')

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/genero\\_candidato/'MASCULINO'](http://127.0.0.1:5000/candidatos_goiás/eleicao/genero_candidato/'MASCULINO')

Etnia:

[http://127.0.0.1:5000/candidatos\\_goiás/candidatos\\_goiás/eleicao/raca\\_candidato/'Parda'](http://127.0.0.1:5000/candidatos_goiás/candidatos_goiás/eleicao/raca_candidato/'Parda')

Existem outras etnias que podem ser pesquisadas, é só substituir no final da url por uma das etnias a seguir:

PARDA

PRETA

BRANCA

Região:

[http://127.0.0.1:5000/candidatos\\_goiás/eleicao/regiao/'LUZIÂNIA'](http://127.0.0.1:5000/candidatos_goiás/eleicao/regiao/'LUZIÂNIA')

onde está escrito 'LUZIÂNIA' entre aspas, substitua por nome de uma região do Goiás