

Projeto

AutoCNPJ

Autor: Caio Lima da Silva
Email: caiolimanagy@protonmail.com
WhatsApp: +55 11 96739-0449

Linkedin: [Caio Lima da Silva | LinkedIn](#)

AutoCNPJ o projeto para mostrar meus conhecimentos consolidados tanto em python quanto em banco de dados, nesse projeto foi utilizado o próprio Python e o binário do postgres para inserção em massa com COPY e priorizando a velocidade, levando em consideração que no brasil tem mais de 55 milhões de empresas, são quase 20gb em dados a baixo vou deixar um dicionário de dos dado e o processo em “passo a passo” de como foi construído e as funcionalidades.

umas das funcionalidades que desenvolvi nesse projeto foi para trabalhar em conjunto com um API em um site de serviços que eu estava construindo,

```
for i in range(3):  
    key = input(f'Digite a chave ({i + 1}/3): ')  
    url = f'https://[REDACTED].com/keys/{key}'  
    # Envia a solicitação HTTP GET para a URL da API  
    response = requests.get(url)  
  
    if response.status_code == 200:  
        try:  
            valid = response.json()['valid']  
            if valid:
```

Uma condição para verificar se o status retornado é o código 200.

- O código HTTP 200 “OK” é uma resposta de status de sucesso que indica que a requisição foi bem-sucedida, ele então ele verifica o JSON retornado, e verifica se contem o STATUS “VALID”, todas as chaves como KEY de acesso, são criados dentro de uma aplicação WEB para permitir a permitir o acesso da aplicação e execução do meu código Python foi todo construído em Laravel um framework PHP para a Web.

Após essa verificação, ele segue realizando a conexão do banco de dados e a testando retornando a versão do postgres e uma mensagem de conexão bem-sucedida e então sendo apresentando um menu, onde exibe as funcionalidades do aplicativo:

0 - Executar todas as opções

- 1 - Baixar arquivos (baixa os arquivos diretamente do servidor da receita federal)
- 2 - Extrair arquivos (extrai os arquivos baixados e após finalizar cada arquivo ele realiza uma limpeza)
- 3 - Criar tabelas (cria todas as tabelas do banco de dados)
- 4 - Importar CSV para o Banco de Dados (após realizar a verificação se não é encontrado nenhum erro em nenhum arquivo ele segue com a importação para o banco de dados)
- 5 - Sair do programa

Todo o processo mencionado acima é apresentado dessa forma a baixo para o usuário:

```
-- AutoCNPJs / Versão: 1.0.0.1
Digite a chave (1/3): SA23NF-42NJG3-MBKDF-326KSF
Chave válida!

Checar conexão com banco de dados...

Digite o nome do banco de dados: postgres
Digite o nome do usuário: postgres
Digite a senha: 140477
Connected to PostgreSQL database: PostgreSQL 15.3, compiled by Visual C++ build 1914, 64-bit
Conexão com o banco de dados estabelecida com sucesso!

0 - Executar todas as opções
1 - Baixar arquivos
2 - Extrair arquivos
3 - Criar tabelas
4 - Importar CSV para o Banco de Dados
5 - Sair do programa
Digite a opção desejada:
```

Começaremos a apresentação por cada funcionalidade:

- Na opção 1: Baixar Arquivos. Utilizando a biblioteca **BeautifulSoup** é feito um **web scraping** para automatizar o download das bases que posteriormente será feito o **IMPORT** para o banco de dados, São mais de 30 Arquivos.

```
Iniciando a busca por arquivos...

Arquivo Cnaes.zip baixado com sucesso!

Arquivo Empresas0.zip baixado com sucesso!

Arquivo Empresas1.zip baixado com sucesso!

Arquivo Empresas2.zip baixado com sucesso!
Downloading Empresas3.zip 37.0/81.2M MB
```

Como está funcionando o código:

bibliotecas utilizadas:

requests: É utilizada para fazer requisições HTTP e obter o conteúdo da página web que contém a lista de arquivos.

BeautifulSoup: É utilizada para analisar o conteúdo HTML e extrair informações dos elementos da página.

Tqdm: Utilizado para criar a barra de progresso

Inicializar a busca por arquivos:

A função imprime "Iniciando a busca por arquivos..." para informar o início da operação.
A URL base do servidor onde os arquivos estão localizados é definida.
Fazer a requisição HTTP para a página que contém a lista de arquivos:

É feita uma requisição GET usando o módulo **requests** para obter o conteúdo da página HTML.

Agora com o **BeautifulSoup**:

O conteúdo da página HTML obtido é analisado usando **BeautifulSoup**, que converte o HTML em uma estrutura de árvore que pode ser facilmente navegada.
Localizar as linhas da tabela HTML que contém informações dos arquivos:

A função usa **soup.find_all("tr")** para localizar todas as tags **<tr>** que representam as linhas da tabela HTML.

As linhas são armazenadas na variável **rows** e então criar um dicionário vazio para armazenar informações dos arquivos:

A variável **arquivos_at** é inicializada como um dicionário vazio, onde serão armazenados os detalhes de cada arquivo encontrado.

Verificar a existência da pasta "downloads" e criá-la se não existir:

O código verifica se a pasta "downloads" já existe no sistema usando **os.path.exists()**.
Se não existir, a função cria essa pasta usando **os.makedirs()**.

Iterar pelas linhas da tabela e extrair informações dos arquivos:

O código percorre cada linha (row) obtida anteriormente.

Em cada linha, ele encontra todas as colunas (cols) usando **row.find_all("td")**.
É verificado se a linha contiver um link válido para um arquivo, as informações do arquivo (nome, link e data de atualização) são extraídas.

Ignorar arquivos indesejados ou links:

Existem algumas verificações condicionais que ignoram determinados arquivos ou links específicos, como "Parent Directory", "LAYOUT_DADOS_ABERTOS.." e "regime_tributario/". Isso garante que apenas arquivos relevantes sejam processados.

Download dos arquivos:

A função começa fazendo o download dos arquivos encontrados na pasta "downloads".

É criado o caminho completo para salvar cada arquivo usando **os.path.join()**.

O arquivo é baixado em pedaços (chunks) para economizar memória e garantir que o processo de download possa ser interrompido e retomado se necessário.

A barra de progresso é exibida durante o download usando **tqdm** para monitorar o progresso do download.

Se ocorrer algum erro ao baixar um arquivo, uma mensagem de erro é exibida, mas o código continua o processo de download dos demais arquivos.

Tempo de espera entre downloads:

Após o download de cada arquivo, a função aguarda 1 segundo (**time.sleep(1)**) antes de iniciar o download do próximo arquivo.

- **Opção 2: Extrair Arquivos**

bibliotecas utilizadas:

os: A biblioteca os é usada para realizar operações relacionadas ao sistema operacional, como manipulação de diretórios e caminhos de arquivos. Ela é usada para construir caminhos para os diretórios de dados e para verificar a existência de diretórios antes de importar os arquivos CSV.

zipfile: Um módulo da biblioteca padrão do Python usado para trabalhar com arquivos ZIP. Ele fornece funcionalidades para extrair arquivos de arquivos ZIP.

logging: Outro módulo nativo do Python que fornece capacidades de registro (logging), ele é usado para registrar quaisquer erros ou informações de depuração relacionadas ao processo de extração.

```
Arquivos baixados com sucesso:  
Extraído 1 de 37 arquivos (2.70%)  
Extraído 2 de 37 arquivos (5.41%)  
Extraído 3 de 37 arquivos (8.11%)  
Extraído 4 de 37 arquivos (10.81%)  
Extraído 5 de 37 arquivos (13.51%)  
Extraído 6 de 37 arquivos (16.22%)  
Extraído 7 de 37 arquivos (18.92%)  
Extraído 8 de 37 arquivos (21.62%)  
Extraído 9 de 37 arquivos (24.32%)  
Extraído 10 de 37 arquivos (27.03%)  
Extraído 11 de 37 arquivos (29.73%)  
Extraído 12 de 37 arquivos (32.43%)  
Extraído 13 de 37 arquivos (35.14%)  
Extraído 14 de 37 arquivos (37.84%)  
Extraído 15 de 37 arquivos (40.54%)  
Extraído 16 de 37 arquivos (43.24%)
```

A função trata o caso em que o diretório "data" não existe e o cria, se necessário e todos os arquivos são colocados dentro da pasta data

A função organiza os arquivos extraídos em diferentes subdiretórios com base nas palavras-chave presentes nos nomes dos arquivos. Cada grupo corresponde a um tipo específico de dado (por exemplo, empresas, estabelecimentos, sócios, etc.).

Se um arquivo ZIP for encontrado corrompido, ele é registrado como um erro com uma mensagem em um log.

O progresso do processo de extração é exibido durante a execução, mostrando quantos arquivos foram extraídos em relação ao total, juntamente com a percentagem concluída.

- **Opção 3: Criar tabelas,** funciona no mesmo formato para todas as tabelas, ele verifica a conexão e caso a tabela exista ele realiza um drop table, faz o commit em seguida executa outro sql utilizando python para criar a tabela novamente no banco de dados.

```
def criar_tabela_motivos(conn):
    cursor = conn.cursor()

    drop_table_query = "DROP TABLE IF EXISTS Motivos;"
    cursor.execute(drop_table_query)
    conn.commit()

    create_table_query = """
        CREATE TABLE Motivos (
            CODIGO int,
            DESCRICAO TEXT
        );
    """
    cursor.execute(create_table_query)
    conn.commit()

    cursor.close()
```

- Na opção 4: Importar CSVs para o Baco de dados.

Bibliotecas utilizadas:

os: A biblioteca os é usada para realizar operações relacionadas ao sistema operacional, como manipulação de diretórios e caminhos de arquivos. Ela é usada para construir caminhos para os diretórios de dados e para verificar a existência de diretórios antes de importar os arquivos CSV.

csv: A biblioteca csv é usada para lidar com a leitura e escrita de arquivos CSV. Ela oferece métodos convenientes para trabalhar com dados estruturados em formato CSV, permitindo que o código leia os dados dos arquivos CSV e obtenha os cabeçalhos e linhas facilmente.

psycopg2: Essa biblioteca é um adaptador de banco de dados PostgreSQL para Python. É utilizada para se conectar e interagir com um banco de dados Postgres. Com ela, é possível executar comandos SQL, realizar consultas e inserir dados no banco de dados.

```
es\Cnaes_0.csv - 1359 linhas copiadas
resas\Empresas9_0.csv - 13834137 linhas copiadas
resas\Empresas9_1.csv - 4494860 linhas copiadas
resas\Empresas9_2.csv - 4494860 linhas copiadas
resas\Empresas9_3.csv - 4494860 linhas copiadas
resas\Empresas9_4.csv - 4494860 linhas copiadas
resas\Empresas9_5.csv - 4494860 linhas copiadas
resas\Empresas9_6.csv - 4494860 linhas copiadas
resas\Empresas9_7.csv - 4494860 linhas copiadas
resas\Empresas9_8.csv - 4494860 linhas copiadas
resas\Empresas9_9.csv - 4494860 linhas copiadas
imentos\Estabelecimentos9_0.csv limpo com sucesso
estabelecimentos\Estabelecimentos9_0.csv - 14375726 linhas copiadas
estabelecimentos\Estabelecimentos9_1.csv - 4753435 linhas copiadas
```

A função `import_csv_to_postgres(conn)` recebe uma conexão ativa com o banco de dados Postgres (`conn`) como argumento.

Um dicionário chamado `table_names` é definido, mapeando nomes de diretórios para nomes de tabelas no banco de dados. Cada diretório representa um tipo de dados (por exemplo, "cnaes" representa Classificação Nacional de Atividades Econômicas).

Em seguida, um loop `for` itera sobre cada diretório no dicionário `table_names`.

Para cada diretório, o código verifica se o caminho do diretório existe. Se não existir, ele pula para o próximo diretório.

Para cada arquivo CSV encontrado no diretório atual, o código realiza algumas operações:

- Remove caracteres NUL do arquivo, se houver, para garantir que os dados não estejam corrompidos.
- Abre o arquivo CSV e lê os dados usando a biblioteca `csv` do Python. Define o delimitador como `;` e o caractere de citação como `"`.
- Obtém os cabeçalhos (nomes das colunas) do arquivo CSV.
- Consulta o banco de dados para obter os nomes das colunas da tabela correspondente ao diretório atual.
- Compara o número de colunas no arquivo CSV com o número de colunas na tabela do banco de dados. Se eles não coincidirem, tenta ajustar o número de colunas adicionando ou removendo colunas vazias no arquivo CSV. Em seguida, insere os dados na tabela correspondente usando o método `copy_expert` do cursor do banco de dados.
- Se o número de colunas for compatível, insere os dados na tabela diretamente usando o método `copy_expert`.

Em caso de erros, o código imprime informações de exceção e realiza um `rollback` dos inserts.

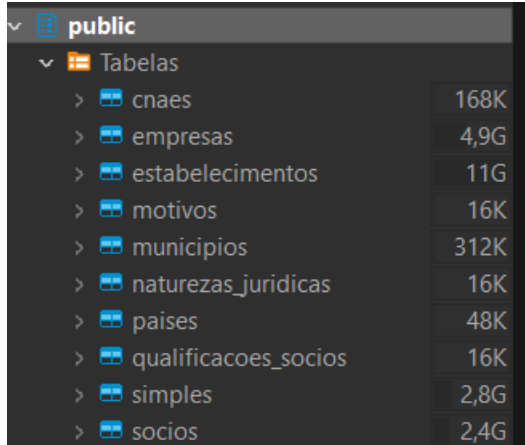
Em resumo, essa funcionalidade lê arquivos CSV de diferentes diretórios e os insere nas tabelas correspondentes do banco de dados Postgres usando a biblioteca psycopg2. Antes da inserção, realiza algumas verificações para garantir que o número de colunas seja compatível entre o arquivo CSV e a tabela do banco de dados.

Todo o processo de import demora cerca de 25 a 30 minutos o que está extremamente otimizado levando em consideração a quantidade de dados. Em outros testes antes de otimizar demorava cerca de 4 a 6 horas.

o resultado final:

Tabela Estabelecimentos:

Nome da coluna	#
ABC cnpj_basico	1
ABC cnpj_ordem	2
ABC cnpj_dv	3
ABC identificador	4
ABC nome_fantasia	5
ABC situacao_cadastral	6
ABC data_situacao_cadastral	7
ABC motivo_situacao_cadastral	8
ABC nome_cidade_exterior	9
ABC pais	10
ABC data_inicio_atividade	11
ABC cnae_fiscal_principal	12
ABC cnae_fiscal_secundária	13
ABC tipo_logradouro	14
ABC logradouro	15
ABC numero	16
ABC complemento	17
ABC bairro	18
ABC cep	19
ABC uf	20



The screenshot shows the 'public' schema in a database interface. Under the 'Tabelas' (Tables) section, a list of tables is displayed with their respective sizes. The 'estabelecimentos' table is highlighted, showing a size of 11G. Other tables include 'cnaes' (168K), 'empresas' (4,9G), 'motivos' (16K), 'municipios' (312K), 'naturezas_juridicas' (16K), 'paises' (48K), 'qualificacoes_socios' (16K), 'simples' (2,8G), and 'socios' (2,4G).

a baixo vou deixar o dicionario dos dados.

EMPRESAS	
Campo	Descrição
CNPJ BÁSICO	NÚMERO BASE DE INSCRIÇÃO NO CNPJ (OITO PRIMEIROS DÍGITOS DO CNPJ).
RAZÃO SOCIAL / NOME EMPRESARIAL	NOME EMPRESARIAL DA PESSOA JURÍDICA
NATUREZA JURÍDICA	CÓDIGO DA NATUREZA JURÍDICA
QUALIFICAÇÃO DO RESPONSÁVEL	QUALIFICAÇÃO DA PESSOA FÍSICA RESPONSÁVEL PELA EMPRESA
CAPITAL SOCIAL DA EMPRESA	CAPITAL SOCIAL DA EMPRESA
PORTE DA EMPRESA	CÓDIGO DO PORTE DA EMPRESA: 00 - NÃO INFORMADO 01 - MICRO EMPRESA 03 - EMPRESA DE PEQUENO PORTE 05 - DEMAIS
ENTE FEDERATIVO RESPONSÁVEL	O ENTE FEDERATIVO RESPONSÁVEL É PREENCHIDO PARA OS CASOS DE ÓRGÃOS E ENTIDADES DO GRUPO DE NATUREZA JURÍDICA 1XXX. PARA AS DEMAIS NATUREZAS, ESTE ATRIBUTO FICA EM BRANCO.

ESTABELECIMENTOS	
Campo	Descrição
CNPJ BÁSICO	NÚMERO BASE DE INSCRIÇÃO NO CNPJ (OITO PRIMEIROS DÍGITOS DO CNPJ).
CNPJ ORDEM	NÚMERO DO ESTABELECIMENTO DE INSCRIÇÃO NO CNPJ (DO NONO ATÉ O DÉCIMO SEGUNDO DÍGITO DO CNPJ).
CNPJ DV	DÍGITO VERIFICADOR DO NÚMERO DE INSCRIÇÃO NO CNPJ (DOIS ÚLTIMOS DÍGITOS DO CNPJ).
IDENTIFICADOR MATRIZ/FILIAL	CÓDIGO DO IDENTIFICADOR MATRIZ/FILIAL: 1 - MATRIZ 2 - FILIAL

NOME FANTASIA	CORRESPONDE AO NOME FANTASIA
SITUAÇÃO CADASTRAL	CÓDIGO DA SITUAÇÃO CADASTRAL: 01 - NULA 2 - ATIVA 3 - SUSPENSA 4 - INAPTA 08 - BAIXADA
DATA SITUAÇÃO CADASTRAL	DATA DO EVENTO DA SITUAÇÃO CADASTRAL

MOTIVO SITUAÇÃO CADASTRAL	CÓDIGO DO MOTIVO DA SITUAÇÃO CADASTRAL
NOME DA CIDADE NO EXTERIOR	NOME DA CIDADE NO EXTERIOR
PAIS	CÓDIGO DO PAIS
DATA DE INÍCIO ATIVIDADE	DATA DE INÍCIO DA ATIVIDADE
CNAE FISCAL PRINCIPAL	CÓDIGO DA ATIVIDADE ECONÔMICA PRINCIPAL DO ESTABELECIMENTO
CNAE FISCAL SECUNDÁRIA	CÓDIGO DA(S) ATIVIDADE(S) ECONÔMICA(S) SECUNDÁRIA(S) DO ESTABELECIMENTO
TIPO DE LOGRADOURO	DESCRIÇÃO DO TIPO DE LOGRADOURO
LOGRADOURO	NOME DO LOGRADOURO ONDE SE LOCALIZA O ESTABELECIMENTO.
NÚMERO	NÚMERO ONDE SE LOCALIZA O ESTABELECIMENTO. QUANDO NÃO HOUVER PREENCHIMENTO DO NÚMERO HAVERÁ 'S/N'.
COMPLEMENTO	COMPLEMENTO PARA O ENDEREÇO DE LOCALIZAÇÃO DO ESTABELECIMENTO
BAIRRO	BAIRRO ONDE SE LOCALIZA O ESTABELECIMENTO.
CEP	CÓDIGO DE ENDEREÇAMENTO POSTAL REFERENTE AO LOGRADOURO NO QUAL O ESTABELECIMENTO ESTA LOCALIZADO
UF	SIGLA DA UNIDADE DA FEDERAÇÃO EM QUE SE ENCONTRA O ESTABELECIMENTO
MUNICÍPIO	CÓDIGO DO MUNICÍPIO DE JURISDIÇÃO ONDE SE ENCONTRA O ESTABELECIMENTO
DDD 1	CONTÉM O DDD 1
TELEFONE 1	CONTÉM O NÚMERO DO TELEFONE 1
DDD 2	CONTÉM O DDD 2
TELEFONE 2	CONTÉM O NÚMERO DO TELEFONE 2
DDD DO FAX	CONTÉM O DDD DO FAX
FAX	CONTÉM O NÚMERO DO FAX
CORREIO ELETRÔNICO	CONTÉM O E-MAIL DO CONTRIBUINTE
SITUAÇÃO ESPECIAL	SITUAÇÃO ESPECIAL DA EMPRESA
DATA DA SITUAÇÃO ESPECIAL	DATA EM QUE A EMPRESA ENTROU EM SITUAÇÃO ESPECIAL

DADOS DO SIMPLES	
Campo	Descrição
CNPJ BÁSICO	NÚMERO BASE DE INSCRIÇÃO NO CNPJ (OITO PRIMEIROS DÍGITOS DO CNPJ).
OPÇÃO PELO SIMPLES	INDICADOR DA EXISTÊNCIA DA OPÇÃO PELO SIMPLES. S - SIM N - NÃO EM BRANCO – OUTROS
DATA DE OPÇÃO PELO SIMPLES	DATA DE OPÇÃO PELO SIMPLES
DATA DE EXCLUSÃO DO SIMPLES	DATA DE EXCLUSÃO DO SIMPLES
OPÇÃO PELO MEI	INDICADOR DA EXISTÊNCIA DA OPÇÃO PELO MEI S - SIM N - NÃO EM BRANCO - OUTROS
DATA DE OPÇÃO PELO MEI	DATA DE OPÇÃO PELO MEI
DATA DE EXCLUSÃO DO MEI	DATA DE EXCLUSÃO DO MEI

SÓCIOS	
Campo	Descrição
CNPJ BÁSICO	NÚMERO BASE DE INSCRIÇÃO NO CNPJ (CADASTRONACIONAL DA
	PESSOA JURÍDICA).
IDENTIFICADOR DE SÓCIO	CÓDIGO DO IDENTIFICADOR DE SÓCIO 1 – PESSOA JURÍDICA 2 – PESSOA FÍSICA 3 – ESTRANGEIRO
NOME DO SÓCIO (NO CASO PF) OU RAZÃO SOCIAL (NO CASO PJ)	NOME DO SÓCIO PESSOA FÍSICA OU A RAZÃO SOCIAL E/OU NOME EMPRESARIAL DA PESSOA JURÍDICA E/OU NOME DOSÓCIO/RAZÃO SOCIAL DO SÓCIO ESTRANGEIRO
CNPJ/CPF DO SÓCIO	CPF OU CNPJ DO SÓCIO (SÓCIO ESTRANGEIRO NÃO TEM ESTA INFORMAÇÃO).
QUALIFICAÇÃO DO SÓCIO	CÓDIGO DA QUALIFICAÇÃO DO SÓCIO
DATA DE ENTRADA NA SOCIEDADE	DATA DE ENTRADA NA SOCIEDADE

PAIS	CÓDIGO PAÍS DO SÓCIO ESTRANGEIRO
REPRESENTANTE LEGAL	NÚMERO DO CPF DO REPRESENTANTE LEGAL
NOME DO REPRESENTANTE	NOME DO REPRESENTANTE LEGAL
QUALIFICAÇÃO DO REPRESENTANTE LEGAL	CÓDIGO DA QUALIFICAÇÃO DO REPRESENTANTE LEGAL
FAIXA ETÁRIA	CÓDIGO CORRESPONDENTE À FAIXA ETÁRIA DO SÓCIO

PAÍSES	
Campo	Descrição
CÓDIGO	CÓDIGO DO PAÍS
DESCRIÇÃO	NOME DO PAÍS

MUNICÍPIOS	
Campo	Descrição
CÓDIGO	CÓDIGO DO MUNICÍPIO
DESCRIÇÃO	NOME DO MUNICÍPIO

QUALIFICAÇÕES DE SÓCIOS	
Campo	Descrição
CÓDIGO	CÓDIGO DA QUALIFICAÇÃO DO SÓCIO
DESCRIÇÃO	NOME DA QUALIFICAÇÃO DO SÓCIO

NATUREZAS JURÍDICAS	
Campo	Descrição
CÓDIGO	CÓDIGO DA NATUREZA JURÍDICA
DESCRIÇÃO	NOME DA NATUREZA JURÍDICA

CNAEs	
Campo	Descrição
CÓDIGO	CÓDIGO DA ATIVIDADE ECONÔMICA
DESCRIÇÃO	NOME DA ATIVIDADE ECONÔMICA