# Compilação de Dados – Importação e Exportação do Estado de São Paulo

**Data de criação:** 15/09/2025  
**Última atualização:** 10/10/2025  
**Autores:** Jackson Penaforte e Vitor Amaral

# Importa a biblioteca necessária  
import pandas as pd  
  
# Monta o Google Drive para acessar os arquivos  
from google.colab import drive  
drive.mount('/content/drive')  
  
# Define o caminho da pasta onde estão os arquivos de dados  
origem = '/content/drive/MyDrive/api\_python/exportcao\_importacao/'  
  
# Define os caminhos dos arquivos CSV  
arq\_7 = origem + 'EXP\_2023\_MUN.csv'  
arq\_8 = origem + 'IMP\_2023\_MUN.csv'  
arq\_9 = origem + 'EXP\_2024\_MUN.csv'  
arq\_10 = origem + 'IMP\_2024\_MUN.csv'  
arq\_11 = origem + 'EXP\_2025\_MUN.csv'  
arq\_12 = origem + 'IMP\_2025\_MUN.csv'  
ncm = origem + 'NCM.csv'  
país = origem + 'PAIS.csv'  
via = origem +'VIA.csv'  
urf = origem + 'URF.csv'  
uf\_m = origem + 'UF\_MUN.csv'  
uf = origem + 'UF.csv'  
sh = origem + 'NCM\_SH.csv'  
  
# Lê os arquivos de exportação e importação municipais  
exp23m = pd.read\_csv(arq\_7, low\_memory=False, sep=';', encoding='UTF-8')  
imp23m = pd.read\_csv(arq\_8, low\_memory=False, sep=';', encoding='UTF-8')  
exp24m = pd.read\_csv(arq\_9, low\_memory=False, sep=';', encoding='UTF-8')  
imp24m = pd.read\_csv(arq\_10, low\_memory=False, sep=';', encoding='UTF-8')  
exp25m = pd.read\_csv(arq\_11, low\_memory=False, sep=';', encoding='UTF-8')  
imp25m = pd.read\_csv(arq\_12, low\_memory=False, sep=';', encoding='UTF-8')  
  
# Lê tabelas auxiliares de referência  
NCM = pd.read\_csv(ncm, low\_memory=False, sep=';', encoding='latin1')  
PAÍS = pd.read\_csv(país, low\_memory=False, sep=';', encoding='latin1')  
VIA = pd.read\_csv(via, low\_memory=False, sep=';', encoding='latin1')  
URF = pd.read\_csv(urf, low\_memory=False, sep=';', encoding='latin1')  
UF\_MUN = pd.read\_csv(uf\_m, low\_memory=False, sep=';', encoding='latin1')  
UF = pd.read\_csv(uf, low\_memory=False, sep=';', encoding='latin1')  
SH = pd.read\_csv(sh, low\_memory=False, sep=';', encoding='latin1')  
  
# Remove linhas duplicadas em tabelas auxiliares  
PAÍS = PAÍS[['CO\_PAIS','NO\_PAIS\_ING']].drop\_duplicates(subset='CO\_PAIS')  
UF\_MUN = UF\_MUN[['CO\_MUN\_GEO','NO\_MUN']].drop\_duplicates(subset='CO\_MUN\_GEO')  
SH = SH[['CO\_SH4','NO\_SH4\_POR']].drop\_duplicates(subset='CO\_SH4')  
  
# Une dados de exportação de 2023 a 2025  
finalexpm = pd.concat([exp23m, exp24m, exp25m], ignore\_index=True)  
  
# Mescla informações de país, município e classificação SH  
finalexpm = finalexpm.merge(PAÍS[['CO\_PAIS','NO\_PAIS\_ING']], on='CO\_PAIS', how='left')  
finalexpm = finalexpm.merge(UF\_MUN[['CO\_MUN\_GEO','NO\_MUN']], left\_on='CO\_MUN', right\_on='CO\_MUN\_GEO', how='left')  
finalexpm = finalexpm.merge(SH[['CO\_SH4','NO\_SH4\_POR']], left\_on='SH4', right\_on='CO\_SH4', how='left')  
  
# Une dados de importação de 2023 a 2025  
finalimpm = pd.concat([imp23m, imp24m, imp25m], ignore\_index=True)  
  
# Mescla informações de país, município e classificação SH  
finalimpm = finalimpm.merge(PAÍS[['CO\_PAIS','NO\_PAIS\_ING']], on='CO\_PAIS', how='left')  
finalimpm = finalimpm.merge(UF\_MUN[['CO\_MUN\_GEO','NO\_MUN']], left\_on='CO\_MUN', right\_on='CO\_MUN\_GEO', how='left')  
finalimpm = finalimpm.merge(SH[['CO\_SH4','NO\_SH4\_POR']], left\_on='SH4', right\_on='CO\_SH4', how='left')  
  
# Filtra dados apenas do estado de São Paulo (SP)  
finalimpSP = finalimpm[finalimpm['SG\_UF\_MUN'] == 'SP']  
finalexpSP = finalexpm[finalexpm['SG\_UF\_MUN'] == 'SP']  
  
# Visualiza as últimas linhas dos dados tratados  
finalexpSP.tail(5)  
finalimpSP.tail(5)  
  
# Exporta os arquivos finais limpos e filtrados por SP  
finalimpSP.to\_csv(origem + 'finalimpSP.csv', index=False)  
finalexpSP.to\_csv(origem + 'finalexpSP.csv', index=False)