

# Versão 01

```
In [169]: import pandas as pd
import openai
```

```
In [170]: df = pd.read_csv('moat_202404.csv')

# Chave da API da OpenAI
openai.api_key = 'sk-ccUgngdqgNvKIphdo9KST3BlbkFJUyiUJ8UMkJ4x0eFLGz'
```

```
In [171]: def traduzir_LN_para_pandas(cmd):

    # passa o prompt para a API da OpenAI
    prompt = f"""
    Eu tenho um DataFrame do pandas e preciso de ajuda para transfo
    para a sintaxe do pandas. Linguagem natural: "{cmd}"
    Sintaxe do pandas: retorne exclusivamente a linha com o comando
    """

    # chama a API da OpenAI para tradução usando o modelo gpt-3.5-t
    resp = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "system", "content": "Você é um assistente úti
            {"role": "user", "content": prompt}
        ],
        max_tokens=100,
        temperature=0.5,
    )

    query_pandas = resp.choices[0].message['content'].strip()
    return query_pandas
```

```
In [172]: # Exemplo de uso: comando de consulta em linguagem natural
cmd_LN = "Mostre os produtos importados cujo valor seja maior que 100000"

# Traduzir o comando para sintaxe pandas
query = traduzir_LN_para_pandas(cmd_LN)

print(f"Comando em linguagem natural: {cmd_LN}")
print(query)
```

Comando em linguagem natural: Mostre os produtos importados cujo valor seja maior que 100000

Claro! Para transformar a consulta em linguagem natural para a sintaxe do pandas, você pode utilizar o seguinte comando:

```
```python
df[(df['Origem'] == 'Importado') & (df['Valor'] > 100000)]
```
```

Esse comando irá retornar exclusivamente as linhas do DataFrame onde a coluna 'Origem' seja igual a 'Importado' e a coluna 'Valor' seja maior que 100000.

```
In [173]: # Executar a consulta no DataFrame
try:
    result = eval(query)
    print(result)
except Exception as e:
    print(f"Erro ao executar a consulta: {str(e)}")
```

Erro ao executar a consulta: invalid syntax (<string>, line 1)

## Versão 02

```
In [166]: import pandas as pd
import re

# Carregar o arquivo CSV em um DataFrame do pandas
file_path = 'moat_202404.csv'
df = pd.read_csv(file_path)

# Dicionário de mapeamento para consulta simples
dic_colunas = {
    'mes': 'anomes',
    'codigo': 'hscodex',
    'identificacao': 'ncm',
    'produto': 'desc_ncm',
    'porto entrada': 'urf_entrada_carga',
    'pais origem': 'nm_pais_origem',
    'cnpj importador': 'importador_cnpj',
    'importador': 'importador_nome',
    'exportador': 'exportador_nome',
    'valor': 'valor_fob',
    'peso': 'peso_liquido'
}
```

```
In [167]: def LN_para_pandas(cmd):

    # este modelo vale apenas para consulta simples de seleção
    match = re.match(r"Mostre (.+) onde (.+)", cmd)
    if not match:
        return "Erro: comando não reconhecido."

    select_part = match.group(1).strip()
    where_part = match.group(2).strip()

    # campos da consulta (SELECT)
    select_fields = []
    for field in select_part.split(","):
        field = field.strip()
        if field in dic_colunas:
            select_fields.append("'" + dic_colunas[field] + "'")
        else:
            return f"Erro: campo desconhecido '{field}'."

    # condições da consulta (WHERE)
    where_conditions = where_part
    for natural, pandas in dic_colunas.items():
        where_conditions = where_conditions.replace(natural, pandas)
    where_conditions = where_conditions.replace(" e ", " & ")
    where_conditions = where_conditions.replace(" ou ", " | ")

    # query pandas final
    select_fields_str = ', '.join(select_fields)
    pandas_query = f"df.query('{where_conditions}')[[{select_fields}]"
    return pandas_query
```

```
In [168]: # Exemplo de uso: comando de consulta em linguagem natural
LN = "Mostre produto, valor onde valor > 100000 e peso < 5000"

# Traduzir o comando para sintaxe pandas
query_pandas = LN_para_pandas(LN)

print(f"Comando em linguagem natural: {LN}")
print(f"Sintaxe pandas: {query_pandas}")

# Executar a consulta no DataFrame
try:
    result = eval(query_pandas)
    print(result)
except Exception as e:
    print(f"Erro ao executar a consulta: {str(e)}")
```

Comando em linguagem natural: Mostre produto, valor onde valor > 100000 e peso < 5000

Sintaxe pandas: df.query('valor\_fob > 100000 & peso\_liquido < 5000')[['desc\_ncm', 'valor\_fob']]

|        | desc_ncm  | valor_f |
|--------|---|---------|
| ob     |   |         |
| 116    | Outras sondas, catéteres e cânulas                | 124213. |
| 00     |   |         |
| 121    | Lentes de outras matérias, para óculos            | 353969. |
| 00     |   |         |
| 140    | Grampos e cliques, seus aplicadores e extratores  | 406079. |
| 00     |   |         |
| 181    | Outras partes para armações de óculos e artigo... | 176331. |
| 00     |   |         |
| 194    | Circuitos impressos multicamadas, com isolante... | 344183. |
| 00     |   |         |
| ...    | ...   |         |
| ...    |   |         |
| 386286 | Outras partes e acessórios para tratores e veí... | 215205. |
| 77     |   |         |
| 386291 | Outras partes e acessórios de carrocerias para... | 131046. |
| 44     |   |         |
| 386352 | Outros aparelhos emissores com receptor incorp... | 292483. |
| 00     |   |         |
| 386358 | Fones de ouvido, mesmo combinados com um micro... | 104171. |
| 00     |   |         |
| 386359 | Outras obras de alumínio                          | 191243. |
| 47     |   |         |

[6841 rows x 2 columns]

