

Nome do Grupo: Agentes Inteligentes Legacy

Alunos:

1. Arthur Neves de Oliveira Santos
2. Fabiana Gonçalves
3. Patrícia Monteiro Barbosa
4. Silvia Marques Almeida Rodrigues

1. Framework

A equipe optou por utilizar o Google Colab para o desenvolvimento da solução. Essa escolha foi influenciada pelas experiências adquiridas no Desafio 2, no qual adotamos uma arquitetura low-code voltada à automação inteligente e à integração de dados em tempo real, utilizando tecnologias como o n8n (plataforma de automação baseada em nós). A partir desse aprendizado, avançamos para uma abordagem baseada exclusivamente em programação em Python.

2. Ferramenta principal usada para o desenvolvimento do projeto:

Para o Desafio 3, a equipe optou por utilizar o Google Colab e a linguagem de programação Python para desenvolver um agente de inteligência artificial voltado à classificação automática de CFOP e NCM. Este projeto tem como objetivo principal revolucionar o compliance fiscal, ao automatizar a classificação de produtos e serviços de forma eficiente, reduzindo erros, otimizando processos e garantindo maior conformidade com a legislação tributária. A iniciativa representa um avanço em relação ao Desafio 2, no qual utilizamos uma arquitetura low-code com foco em automação inteligente e integração de dados.

Agente Inteligente para Notas Fiscais - Google Colab Este notebook implementa um agente inteligente capaz de ler um arquivo CSV com dados de Notas Fiscais e responder perguntas em linguagem natural sobre ele.

🔧 Passo 1 — Instalar bibliotecas necessárias(OpenAI, langchain, pandas)

```
[1] !pip install openai langchain pandas
!pip install langchain_experimental
!pip install -U langchain-openai
```

🔧 Passo 2 — Importar bibliotecas

```
import os
os.environ["OPENAI_API_KEY"] = "INSIRE A CHAVE AQUI"
```

```
[ ] import pandas as pd
from langchain import OpenAI
from langchain_experimental.agents.agent_toolkits.csv.base import create_csv_agent
```

```
[ ] import os
import pandas as pd
from langchain_experimental.agents import create_csv_agent
from langchain.llms import OpenAI
```

🚩 Passo 3 — Conectar ao Google Drive

```
from google.colab import drive
drive.mount('/content/drive')
```

🚩 Passo 4 — Descompactar arquivo ZIP

1. Item da lista
2. Item da lista

Agora, use o código para descompactar o arquivo ZIP. Lembre-se de substituir `'/content/drive/My Drive/caminho/para/seu_arquivo.zip'` pelo caminho real do seu arquivo no Google Drive e `'/content/drive/My Drive/caminho/para/extrair'` pelo diretório de destino desejado (também no seu Drive ou em outro local acessível).

```
[ ] import zipfile
import os

# Substitua pelo caminho completo do seu arquivo .zip no Google Drive
zip_file_path = '/content/drive/My Drive/Colab Notebooks/202401_NFs_Desafio_3.zip'

# Substitua pelo diretório onde você quer extrair os arquivos (pode ser no Drive ou em outro lugar)
extract_dir = '/content/drive/My Drive/Colab Notebooks/NotasFiscaisDescompactadas'

# Cria o diretório de destino se ele não existir
os.makedirs(extract_dir, exist_ok=True)

try:
    with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
        zip_ref.extractall(extract_dir)
        print(f"Arquivo '{zip_file_path}' descompactado com sucesso em '{extract_dir}'.")
except FileNotFoundError:
    print(f"Erro: Arquivo '{zip_file_path}' não encontrado. Verifique o caminho.")
except zipfile.BadZipFile:
    print(f"Erro: O arquivo '{zip_file_path}' não é um arquivo ZIP válido.")
except Exception as e:
    print(f"Ocorreu um erro durante a descompactação: {e}")
```

🚩 Passo 5 — Ler os CSVs

```
[ ] import pandas as pd

# Caminhos para os dois arquivos
cabecalho_path = '/content/drive/MyDrive/Colab Notebooks/NotasFiscaisDescompactadas/202401_NFs_Cabecalho.csv'
itens_path = '/content/drive/MyDrive/Colab Notebooks/NotasFiscaisDescompactadas/202401_NFs_Itens.csv'

# Lendo os CSVs
df_cabecalho = pd.read_csv(cabecalho_path)
df_itens = pd.read_csv(itens_path)

# Mostrando as primeiras linhas
print("\n Cabecalho:")
display(df_cabecalho.head())

print("\n Itens:")
display(df_itens.head())
```

🚀 Passo 6 — Criando Agente para o Cabeçalho 🤖

```
[ ] from langchain_openai import ChatOpenAI
    from langchain_experimental.agents import create_pandas_dataframe_agent

    llm = ChatOpenAI(
        model="mistralai/mistral-7b-instruct", # ou outro modelo suportado
        temperature=0,
        api_key="INSIRA A CHAVE AQUI",
        base_url="https://openrouter.ai/api/v1"
    )

    agent_cabecalho = create_pandas_dataframe_agent(
        llm,
        df_cabecalho,
        verbose=True,
        allow_dangerous_code=True,
        handle_parsing_errors=True
    )
    agent_cabecalho.invoke("Quantas notas fiscais foram emitidas?")
```


🚀 Passo 7 — Criando Agente para os Itens 🤖

```
[ ] from langchain.prompts import PromptTemplate


    # Prompt customizado
    custom_prefix = """
    Você é um agente especialista em análise de notas fiscais.
    Sempre que perguntado sobre produtos mais vendidos, você deve procurar o valor máximo na coluna QUANTIDADE
    e devolver a DESCRIÇÃO correspondente a essa linha no dataframe.
    Responda em português e de forma objetiva.
    """

    agent_itens = create_pandas_dataframe_agent(
        llm,
        df_itens,
        verbose=True,
        allow_dangerous_code=True,
        prefix=custom_prefix
    )
    agent_itens.invoke("Qual foi o produto mais vendido?")
```

3. Perguntas e as Respectivas Respostas

Agente para o a planilha que contém o Cabeçalho 

```
> Entering new AgentExecutor chain...  
  Thought: To find the number of rows in the dataframe, which represents the number of notas fiscais emitidas.  
  Action: python_repl_ast  
  Action Input: `len(df)`  
  Final Answer: 100 notas fiscais were emitted.  
  
> Finished chain.  
{'input': 'Quantas notas fiscais foram emitidas?',  
  'output': '100 notas fiscais were emitted.'}
```

Agente para o a planilha que contém os Itens 

```
> Entering new AgentExecutor chain...  
  Thought: To find the product that was sold the most, I need to find the maximum value in the 'QUANTIDADE' column and then  
  
  Action: python_repl_ast  
  Action Input: `df['QUANTIDADE'].max()`  
  The maximum value in the 'QUANTIDADE' column is 51000.0. Now, I need to find  
  
  Action: python_repl_ast  
  Action Input: `df[df['QUANTIDADE'] == 51000]['DESCRIÇÃO DO PRODUTO/SERVIÇO'].iloc[0]`  
  DIPIFARMA INJETAVEL(DIPIRONA MONOIDR 5
```

4. Link para o Repositório do Github:

https://github.com/Patricia-Barbosa/Agente_NF/tree/main/python/nfe