**Projeto Final**

*CODERHOUSE*

*Curso*: Python - *Tuma*: 63730

*Professor*: Gabriel Rodrigues

Aluno: Francis Martins Vieira

Turma: 63730 – PYTHON

*CODERHOUSE*

Construção pipeline de dados

*Origem: APIs dados abertos IBGE*

1. **Controle de versão**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Responsável** |
| 10/12/2024 | 1.0 | Criação documento | Francis Martins Vieira |
| 17/10/2024 | 1.1 | Revisão e ajuste para documento em versão “Entrega Final” | Francis Martins Vieira |
|  |  |  |  |

1. **Introdução**

**2.1. Objetivo**

Este projeto tem como objetivo a construção de um código que coleta dados de 3 APIs, relacionadas a dados abertos do IBGE, e após atividade de processamento e integração dos dados seja gerado um banco de dados consolidado que viabilize analisar informações de frequência de registro de Nomes de pessoas no país.

**2.2. Público-alvo**

O código deste projeto destina-se principalmente a analista de dados que tenham interesse em estudos relacionado a frequência de registro de Nomes de pessoas no país, mas podendo ser possível sua utilização por desenvolvedores a fim de consultas e/ou estudo.

**2.3. Saída**

Ao final do código foi acrescentado instrução para exportação das tabelas do banco de dados em arquivos csv, com estrutura a seguir:

TB\_Estados.csv, campos: [id,sigla,nome,regiao\_id,regiao\_sigla,regiao\_nome]

TB\_Rank\_Nomes.csv, campos: [localidade\_id,nome,frequencia,ranking]

TB\_Nome\_Decada.csv, campos: [nome,localidade,frequencia 1930[,"frequencia [1930,1940[","frequencia [1940,1950[","frequencia [1950,1960[","frequencia [1960,1970[","frequencia [1970,1980[","frequencia [1980,1990[","frequencia [1990,2000[","frequencia [2000,2010["]

**2.4. Nível de Privacidade**

As informações extraídas estão disponibilizadas em APIs de dados abertos diante da subjetividade e imparcialidade dos dados, e, por isso, não foram utilizadas soluções de segurança que visem garantir privacidade neste projeto.

1. **Pré-requisitos**

**3.1. Ambientes**

Neste projeto foi utilizado o Visual Studio Code, versão 1.96.0 (user setup) e Python versão 3.13.1 em ambiente com Sistema Operacional Windows\_NT x64 versão 10.0.19045.

**3.2 Bibliotecas**

As bibliotecas utilizadas e necessárias para execução do código são:

* ***notification***, versão 0.2.1;
* ***plyer***, versão 2.1.0;
* ***pandas***, versão 2.2.3;
* ***requests***, versão 2.32.3;
* ***missingno***, versão 0.5.2;

**3.3 Arquivos**

Para execução deste projeto é necessário o arquivo “*requirements.txt*” em mesmo diretório de execução do arquivo de código fonte.

E com execução de todo código serão disponibilizados os arquivos de saída mencionado no item 2.3. Saída.

1. **APIs utilizadas**
2. **API**: v1/localidades/estados
   1. **URL**: https://servicodados.ibge.gov.br/api/v1/localidades/estados
   2. **Utilização**: chamada da URL completa apenas, não há informação adicional necessária para execução.
   3. **Descrição**: API compõem acervo do serviço de dados abertos IBGE de localidades, onde conta um conjunto de APIs referente as divisões de mesorregiões, microrregiões e às divisões político-administrativas do Brasil. Campos: "id", "sigla", "nome", "regiao" onde "regiao" conta lista com {"id", "sigla", "nome"} de cada um dos estados
3. **API**: api/v2/censos/nomes/ranking?localidade=*{id}*
   1. **URL**: https://servicodados.ibge.gov.br/api/v2/censos/nomes/ranking?localidade=*{id}*
   2. **Utilização**: onde está informado a variável {id} na URL de acesso deve ser alterado por um *inteiro* com indetify da localidade, informação obtida na API anterior “v1/localidades/estados”
   3. **Descrição**: API compõem acervo do serviço de dados abertos IBGE, utiliza como base o Censo Demográfico 2010 do IBGE, tem como propósito apresentar um ranking de Top 20 de Nomes por frequência da localidade informada pelo {id}. Campos: "localidade", "sexo" e "res" onde "res" conta lista com *{"nome","frequencia","ranking":1}, [...], {"nome","frequencia","ranking":20}*
4. **API**: v2/censos/nomes/{nome}
   1. **URL**: https://servicodados.ibge.gov.br/api/v2/censos/nomes/*{nome}*
   2. **Utilização**: onde está informado a variável {nome} na URL de acesso deve ser alterado por uma *string* de nome, exemplo: MARIA
   3. **Descrição**: API compõem acervo do serviço de dados abertos IBGE, utiliza como base o Censo Demográfico 2010 do IBGE, tem como propósito apresentar a frequência no Nome no país ao longo das décadas, de 1930 a 2010. Campos: "nome", "sexo", "localidade", "res" onde "res" consta lista com *"periodo":"1930[", "periodo":"[1930,1940[", "periodo":"[1940,1950[", {"periodo":"[1950,1960[", {"periodo":"[1960,1970[", {"periodo":"[1970,1980[", {"periodo":"[1980,1990[", {"periodo":"[1990,2000[" e {"periodo":"[2000,2010["*
5. **Funções criadas**
6. **Nome**: Alerta\_default
   1. **Parâmetros**: *string* com código retorno
   2. **Retorno**: apresentada mensagem de erro com notificação em tela da chamada da API devido *status\_code* recebido ser diferente de 200.
   3. **Descrição**: função tem objetivo de gerar alerta para etapa de extração das informações de retorno da API.
7. **Nome**: buscar\_nomes
   1. **Parâmetros**: *string* com informação de nome
   2. **Retorno**: faz chamadas para API api/v2/censos/nomes/ utilizando função drop\_duplicates que desconsidera informações duplicadas obtidas.
   3. **Descrição**: função tem objetivo de obter os dados da API v2/censos/nomes/{nome} para cada ocorrência única de nome, garantido a remoção de duplicados.
8. **Tratamentos aplicados**

**6.1. Erros**

Erro chamada API: Foram considerados possíveis erros no retorno de chamada das APIs e, para tanto, foi criada função de notificação de Alerta: Alerta\_default.

**6.2. Limpeza de dados**

Utilizada biblioteca *missingno* para verificar se há ocorrência de campos com valores nulos e selecionado para essa finalidade, devido a praticidade de visualização. E em caso afirmativo, em que toda coluna está nula, realizada ação de exclusão.

1. **Banco de dados**

Utilizando biblioteca sqlite3 foram criadas e schema de tabelas para gravação das informações em banco de dados.

Após ação, foram realizadas consultas em tabela do banco de dados para validar a efetividade da criação e analisar informações via consultas. Exemplo:

#Consultando TB\_Estado com "condição” para exibir os dados filtrados pela região "Norte"

# Conectando ao banco de dados

conn = sqlite3.connect('BD\_IBGE\_Nomes.db')

# consulta tabela 'TB\_Estado'

consulta\_TB\_Estado = '''SELECT  id ID\_Estado,

                                nome Estado,

                                sigla UF,

                                regiao\_nome 'Região'

                        FROM TB\_Estado

                        WHERE regiao\_nome = 'Norte' '''

# Gravando informação em variável temporária

temp\_Estado = pd.read\_sql\_query(consulta\_TB\_Estado, conn)

# Fechando a conexão com o banco de dados

conn.close()

#apresentando o DataFrame

temp\_Estado

1. **Versionamento**

Com base no arquivo “*requirements.txt*” segue lista de bibliotecas e suas versões utilizados neste projeto:

asttokens==3.0.0

certifi==2024.8.30

charset-normalizer==3.4.0

colorama==0.4.6

comm==0.2.2

contourpy==1.3.1

cycler==0.12.1

debugpy==1.8.9

decorator==5.1.1

executing==2.1.0

fonttools==4.55.3

idna==3.10

ipykernel==6.29.5

ipython==8.30.0

jedi==0.19.2

jupyter\_client==8.6.3

jupyter\_core==5.7.2

kiwisolver==1.4.7

matplotlib==3.9.3

matplotlib-inline==0.1.7

missingno==0.5.2

nest-asyncio==1.6.0

notification==0.2.1

numpy==2.2.0

packaging==24.2

pandas==2.2.3

parso==0.8.4

pillow==11.0.0

platformdirs==4.3.6

plyer==2.1.0

prompt\_toolkit==3.0.48

psutil==6.1.0

pure\_eval==0.2.3

Pygments==2.18.0

pyparsing==3.2.0

python-dateutil==2.9.0.post0

pytz==2024.2

pywin32==308

pyzmq==26.2.0

requests==2.32.3

scipy==1.14.1

seaborn==0.13.2

six==1.17.0

stack-data==0.6.3

tornado==6.4.2

traitlets==5.14.3

typing\_extensions==4.12.2

tzdata==2024.2

urllib3==2.2.3

wcwidth==0.2.13

1. **Referências**

Documentação APIs: <https://servicodados.ibge.gov.br/api/docs/>

IBGE, Censo Demográfico 2010: <https://censo2010.ibge.gov.br/nomes/#/search>

Doc API localidades: <https://servicodados.ibge.gov.br/api/docs/localidades>

Doc API Nomes: <https://servicodados.ibge.gov.br/api/docs/nomes?versao=2>