**Projeto de Integração de API e Geração de Banco de Dados**

Autor: Loan Alencar Schlemmer

Data: 14 de Dezembro de 2024

**1. Introdução**

**Objetivo:**

Este projeto tem como objetivo coletar dados da API Rest Countries, processá-los e gerar um banco de dados consolidado para análise de informações relacionadas a países, como continentes, moedas e idiomas. O código automatiza as etapas de extração, tratamento e armazenamento dos dados para facilitar o acesso a informações relevantes.

**Público-Alvo:**

Este código é destinado a analistas de dados, desenvolvedores ou interessados em utilizar informações consolidadas de países para fins de análise ou relatórios.

**Saída:**

O banco de dados gerado contém três tabelas principais:

* Continentes: Relaciona países aos seus continentes.
* Moedas: Relaciona países às suas moedas principais.
* Idiomas: Relaciona países aos idiomas falados.

**Nível de Privacidade:**

Os dados coletados são públicos, provenientes de uma API aberta. Nenhuma informação sensível é manipulada.

**2. Pré-Requisitos**

**Ambientes:**

Python 3.9 ou superior.

Visual Studio Code 1.96.0

**Bibliotecas:**

* pandas
* requests
* sqlite3
* plyer

**Arquivos Necessários:**

Código Python (extensão ipynb) e o acesso à API.

**3. APIs Utilizadas**

**Nome da API:** Rest Countries

* Descrição: A API fornece informações detalhadas sobre países, incluindo população, idiomas, moedas e localização geográfica.
* Documentação: https://restcountries.com/

**4. Bibliotecas Utilizadas**

* pandas: Manipulação e limpeza de dados.
* requests: Extração de dados via API.
* sqlite3: Armazenamento dos dados em um banco relacional.
* plyer: Sistema de notificações para alertas.

**5. Funções Criadas**

* extrair\_dados
  + Parâmetros: Nenhum.
  + Retorno: DataFrame com os dados brutos extraídos da API.
  + Descrição: Realiza a conexão com a API Rest Countries e retorna os dados em formato JSON processado.
* tratar\_dados
  + Parâmetros: DataFrame bruto da API.
  + Retorno: Três DataFrames (continentes, moedas, idiomas).
  + Descrição: Limpa e transforma os dados brutos, criando tabelas específicas para cada tipo de informação.
* carregar\_banco
  + Parâmetros: DataFrames tratados e nome do banco de dados.
  + Retorno: Nenhum.
  + Descrição: Salva os dados tratados em um banco de dados SQLite.
* validar\_dados
  + Parâmetros: Nome do banco de dados.
  + Retorno: Nenhum.
  + Descrição: Verifica a integridade dos dados no banco, listando a quantidade de registros por tabela.

**6. Tratamentos Aplicados**

* **Erros:** 
  + Tratamento de erros de conexão com a API, incluindo tentativas de reconexão.
  + Tratamento de falhas na escrita no banco de dados com logs detalhados para diagnóstico.
* **Limpeza de Dados:**
  + Substitui valores ausentes em população por 0.
  + Normaliza nomes de colunas.
  + Remove duplicatas.
  + Gera tabelas estruturadas para continentes, moedas e idiomas.

**7. Método de Saída**

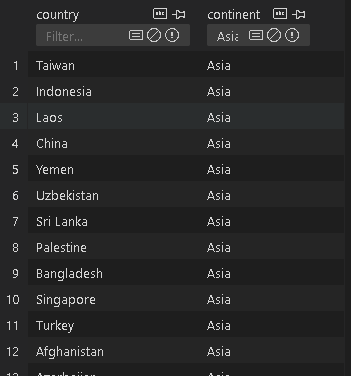
* Formato: Banco de dados SQLite.
* Estrutura:
  + Tabela 1: Continentes
    - Campos: country (string), continent (string).
  + Tabela 2: Moedas
    - Campos: country (string), currency (string).
  + Tabela 3: Idiomas
    - Campos: country (string), language (string).
* Persistência: O banco de dados gerado é salvo localmente no arquivo dados\_tratados.db, que pode ser utilizado diretamente em ferramentas analíticas ou exportado para outros formatos.

**8. Exemplo de Consulta**

Consulta: Listar todos os países de um determinado continente.

SQL: SELECT country FROM continentes WHERE continent = 'Asia';

Resultado:



Consulta Alternativa: Verificar as moedas dos países na Europa.

SQL: SELECT country, currency FROM moedas WHERE country IN (

SELECT country FROM continentes WHERE continent = 'Europe');

**9. Versionamento**

As dependências do projeto estão descritas no arquivo requirements.txt:

* pandas==1.5.0
* requests==2.28.0
* sqlite3==3.36.0
* plyer==2.0.0

**10. Referências**

* Rest Countries API: <https://restcountries.com/>
* Documentação do Pandas: <https://pandas.pydata.org/docs/>
* SQLite Documentation: <https://www.sqlite.org/docs.html>
* Plyer Documentation: <https://plyer.readthedocs.io/en/latest/>