**Documentação do Código: Integração de APIs de Índices Econômicos e Geração de Banco de Dados**

Sillas dos Santos Silva

Atualizado em 06/09/2024

Sumário

[1. Introdução 1](#_Toc176542566)

[2. Pré-requisitos 1](#_Toc176542567)

[3. APIs Utilizadas 1](#_Toc176542568)

[4. Funções Criadas 2](#_Toc176542569)

[5. Tratamentos Aplicados 2](#_Toc176542570)

[6. Método de Saída 2](#_Toc176542571)

[7. Exemplo de Consulta 3](#_Toc176542572)

[8. Referências 3](#_Toc176542573)

### 

### 1. Introdução

* Objetivo: este código tem como objetivo coletar dados das APIs de Selic, IPCA e IGP-M, processá-los e gerar um banco de dados para análise temporal desses índices econômicos do Brasil, assim como análises correlacionais e financeiras.
* Público-alvo: este código é destinado a analista de dados que desejam realizar análises financeiras e macroeconômicas do Brasil com base nos índices básicos de Selic, IPCA e IGP-M
* Saída: a saída será uma tabela em um banco de dados com os campos de data (relativo a cada mês), o valor da Selic (acumulada no mês e anualizada), o valor do IPCA registrado no mês, o IPCA acumulado dos últimos 12 meses, o valor do IGP-m registrado no mês e o IGP-M acumulado dos últimos 12 meses.
* Nível de privacidade: os dados coletados são abertos do Banco Central do Brasil, consultados em <https://dadosabertos.bcb.gov.br/>

### 2. Pré-requisitos

* Ambientes:
  + Versão do Python: Python 3.12.4
* Bibliotecas:
  + pandas 2.2.2
  + requests 2.32.3
  + sqlite3 2.0.32
  + numpy 2.1.0
  + plyer 2.1.0
  + datetime
  + dateutil

### 3. APIs Utilizadas

* Nome da API 1: Taxa de juros - Selic acumulada no mês anualizada
* Descrição breve: API que fornece 2 colunas, sendo de data e o valor da Selic acumulada em cada mês e anualizada
* Nome da API 2: Índice Nacional de Preços ao Consumidor-Amplo (IPCA)
* Descrição breve: API que fornece 2 colunas, sendo de data e o valor do IPCA registrado no mês
* Nome da API 3: Índice Geral de Preços Mercado (IGP-M)
* Descrição breve: API que fornece 2 colunas, sendo de data e o valor IGP-M em cada mês

### 4. Funções Criadas

1. Função: alerta
   1. Parâmetros: recebe os parâmetros nivel, base, etapa e erro
   2. Retorno: retorna uma notificação com dados do erro
   3. Descrição: função que gera um alerta de erro com informação do tipo, base, etapa e horário do erro
2. Função: get\_json\_api
   1. Parâmetros: recebe uma URL
   2. Retorno: retorna o JSON de saída da API da URL
   3. Descrição: função de requisição GET, a partir de uma URL retorna o JSON de saída da API
3. Funções: get\_selic, get\_ipca, get\_igpm
   1. Parâmetros: data\_inicial e data\_final
   2. Retorno: retorna o DataFrame do JSON de cada API
   3. Descrição: a partir da função get\_json\_api e das datas iniciais e finais, que são utilizadas como filtros da API de saída, retorna o DataFrame de cada tabela de dados
4. Função: tabelas\_bd
   1. Parâmetros: NA
   2. Retorno: retorna o DataFrame com as tabelas do banco de dados
   3. Descrição: faz a conexão com o banco e executa uma query para trazer o DataFrame com as tabelas do banco de dados
5. Função: salva\_bd
   1. Parâmetros: recebe um dataframe df e uma string nome\_tabela
   2. Descrição: faz a conexão com o banco e salva o df na tabela ‘nome\_tabela’
6. Função: salva\_bd
   1. Parâmetros: recebe uma tabela nome\_tabela
   2. Descrição: faz a conexão com o banco e carrega a tabela nome\_tabela em um dataframe
7. Função: etapa\_extracao
   1. Parâmetros: data\_inicial e data\_final
   2. Descrição: extrai os dados das APIs selecionadas e salva no banco de dados. Caso falhe na requisição, entra na função alerta
8. Função: etapa\_transformação
   1. Parâmetros: NA
   2. Retorno: indica que salvou a tabela\_indices
   3. Descrição: carregas as tabelas das APIs, faz alguns tratamentos nos campos, cria as colunas de IPCA e IGP-M acumulados em 12 meses e, por fim, faz o merge das 3 tabelas a partir da coluna ‘data’. Em caso de erro, entra na função de alerta

### 5. Tratamentos Aplicados

* Rename: foi feito o rename das colunas valor de cada tabela, de modo que cada valor fosse nomeado na forma valor\_{nome do índice}
* Converte: as colunas de valores dos índices foram convertidas para float para poder aplicar fórmulas numéricas
* Criação de colunas: para o IPCA e o IGP-M foram criadas colunas com esses valores acumulados em 12 meses
* Merge: os dados das 3 APIs foram unidos a partir do campo data, comum nas três tabelas

### 6. Método de Saída

* Formato: banco de dados
* Estrutura: tabela com os campos data (object), valor\_selic (object), valor\_ipca (float), ipca\_acum\_12 (float), valor\_igpm (float), igpm\_acum\_12 (float)

### 7. Exemplo de Consulta

* Consulta:
  + SELECT \* FROM ‘tabela\_indices’
* Resultado:



### 8. Referências

* APIs:
  1. Selic: [https://api.bcb.gov.br/dados/serie/bcdata.sgs.4189/dados?formato=json&dataInicial={data\_inicial}&dataFinal={data\_final](https://api.bcb.gov.br/dados/serie/bcdata.sgs.4189/dados?formato=json&dataInicial=%7Bdata_inicial%7D&dataFinal=%7Bdata_final)}
  2. IPCA: [http://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados?formato=json&dataInicial={data\_inicial}&dataFinal={data\_final](http://api.bcb.gov.br/dados/serie/bcdata.sgs.433/dados?formato=json&dataInicial=%7Bdata_inicial%7D&dataFinal=%7Bdata_final)}
  3. IGP-M: [https://api.bcb.gov.br/dados/serie/bcdata.sgs.189/dados?formato=json&dataInicial={data\_inicial}&dataFinal={data\_final](https://api.bcb.gov.br/dados/serie/bcdata.sgs.189/dados?formato=json&dataInicial=%7Bdata_inicial%7D&dataFinal=%7Bdata_final)}
* APIs consultadas em <https://dadosabertos.bcb.gov.br/>