# Projeto Inteligência de Mercado

**Documentação Técnica Detalhada**

## 1. Introdução

Este projeto visa estruturar uma base técnica para correlação entre dados macroeconômicos e os indicadores internos de produção de caminhões (build rate e mix), fornecendo subsídios confiáveis para decisões estratégicas. O foco está na coleta automatizada, padronização de dados e flexibilidade analítica.

## 2. Arquitetura do Projeto

* buildrate/: pacotes e scripts de coleta automatizada, organizados por tema
* data/: armazenamento dos dados CSV por domínio (ex: transport, macro, credit, exchange\_rates)
* utils/: funções reutilizáveis como o `sgs\_downloader`
* runner.py: script principal que executa todos os coletores via subprocesso
* logs/: histórico das execuções (em versões futuras com timestamps e erros)
* notebooks/: espaço de exploração e validação de dados
* reports/: relatórios gerados automaticamente
* models/: espaço reservado para modelos de previsão e análise futura

## 3. Scripts de Coleta Implementados

* collect\_agropecuaria.py → PIB Agropecuário
* collect\_consumo.py → Comércio varejista ampliado
* collect\_exportacoes.py → (em análise, erro 504 SGS)
* collect\_ibcbr.py → Atividade Econômica (IBC-Br)
* collect\_PIB.py → Produto Interno Bruto
* collect\_producao\_industrial.py → Indústria Transformadora
* collect\_ipca.py / collect\_igpm.py → Inflação
* collect\_selic.py / collect\_cdi.py → Juros
* collect\_usd\_brl.py / collect\_eur\_brl.py → Câmbio
* collect\_credito\_empresas.py → Crédito PJ (SGS 20616)
* collect\_brent.py → Preço do petróleo Brent (EIA)
* collect\_free\_ontl.py → Preço médio do frete rodoviário (R$/TKU via ONTL)
* collect\_unemployment\_pnadc\_ipea.py → Taxa de Desemprego (IPEA via ipeadatapy)

## 4. Execução Automatizada

O arquivo runner.py executa todos os scripts automaticamente. Ele utiliza subprocessos que respeitam o ambiente virtual (via sys.executable) e executa scripts no formato:

**python -m buildrate.collection.collect\_<script>**

Esse padrão garante compatibilidade com o uso de pacotes e importações relativas.

## 5. Funcionamento Interno do runner.py

* Define lista de scripts a serem executados sequencialmente.
* Executa cada um com subprocess.run() e o Python ativo do venv.
* Opcionalmente gera log em CSV com timestamp, nome do script e status.
* Exibe mensagem final no terminal com erro/sucesso de cada script.

## 6. Estrutura dos Dados Coletados

Cada CSV gerado segue a seguinte estrutura de colunas:

* code → código da série (ex: SGS20616, PNADC12\_TDESOCM12)
* raw\_date → data original no formato datetime
* day / month / year → colunas derivadas para filtros temporais
* value → valor numérico da observação

## 7. Séries com Erros de API

* Exportações de Bens (SGS 22663) → Erro 504
* Exportações FOB (SGS 22601) → Erro 504
* Desemprego SGS 24369 → substituído por IPEA PNADC12\_TDESOCM12

## 8. Próximas Etapas Técnicas

* Criar camada de versionamento de dados (por data de coleta)
* Desacoplar coleta e transformação com pipelines separados
* Incluir logs estruturados por script com data/hora e tempo de execução
* Expandir integração com bases IBGE/SIDRA para setores econômicos específicos
* Implementar testes automatizados nos scripts de coleta