# Projeto Inteligência de Mercado

**Visão Estratégica e Técnica para Diretores**

## 1. Visão Geral

Este projeto foi concebido para antecipar cenários econômicos que impactam o planejamento de produção e o portfólio de produtos da montadora. Por meio da correlação entre dados macroeconômicos e variáveis internas de produção (como build rate e mix), fornecemos uma base analítica sólida para a alta gestão.

## 2. Objetivos Estratégicos

* • Antecipar movimentos econômicos que afetam o setor de caminhões.
* • Simular cenários de mercado com base em dados históricos.
* • Fornecer dashboards e relatórios para tomada de decisão executiva.
* • Acompanhar segmentos-chave da economia (agro, bebidas, madeira etc.).
* • Permitir análises orientadas por linguagem natural no futuro.

## 3. Fontes de Dados

* • Banco Central do Brasil (API SGS)
* • IPEA (via ipeadatapy)
* • IBGE/SIDRA (futuro próximo)
* • ONTL (frete rodoviário)
* • ANP (preço do diesel)

## 4. Estrutura Técnica

* • buildrate/: scripts de coleta e tratamento de dados
* • data/: arquivos CSV organizados por domínio
* • utils/: funções reutilizáveis (ex: sgs\_downloader)
* • notebooks/: análises exploratórias e testes de modelo
* • reports/: relatórios gerados automaticamente
* • runner.py: orquestrador principal de coleta de dados

## 5. Indicadores Econômicos Monitorados

* • Câmbio: USD/BRL (SGS 1), EUR/BRL (SGS 21619)
* • Inflação e Juros: IPCA (SGS 433), Selic (SGS 1178), CDI (SGS 12), IGP-M (SGS 189)
* • Atividade Econômica: IBC-Br (SGS 24363), PIB (SGS 4380), PIB Agro (SGS 4385), Consumo (SGS 4382), Indústria (SGS 21859)
* • Crédito: Crédito PJ (SGS 20616)
* • Frete: Média R$/TKU (ONTL)
* • Emprego: Taxa de Desemprego PNAD Contínua (PNADC12\_TDESOCM12 - IPEA)

## 6. Séries com Problemas de Coleta

* • Exportações FOB (SGS 22601) — Erro 504 na API
* • Exportações de Bens (SGS 22663) — Erro 504
* • Desemprego via SGS 24369 — Substituído por série IPEA

## 7. Automação com runner.py

O projeto utiliza um orquestrador central (`runner.py`) que executa todos os scripts de coleta com o comando:

**python runner.py**

O runner utiliza subprocessos com base no Python ativo (venv) e garante que todos os dados sejam atualizados corretamente. Os scripts são executados no padrão `python -m buildrate.collection.<script>`, o que evita erros de importação.

## 8. Próximos Passos

* • 1. Integrar dados internos da montadora para enriquecer análises de build rate e mix.
* • 2. Incorporar fontes complementares (FGV, CEPEA, ANTT).
* • 3. Criar visualizações executivas no Power BI/Tableau com foco em decisão.
* • 4. Desenvolver mecanismo de input textual para simulação de cenários.
* • 5. Migrar armazenamento de dados para SQL Server ou Snowflake com versionamento automatizado.