# APLICAÇÃO DE DESIGN DE SOFTWARE + DIAGRAMA UML DO PROJETO

## Dashboard Interativo com Foco em Importação e Exportação – Previsões Econômicas

Este documento apresenta a aplicação dos princípios de design de software e um diagrama UML para o projeto de dashboard de análise econômica com foco em importação e exportação. A solução permite a visualização de dados históricos de indicadores econômicos obtidos via API do Banco Central, além de aplicar modelos de predição para antecipar variações que impactam decisões comerciais internacionais.

## Design de Software Aplicado

O projeto segue os princípios de orientação a objetos e separação de responsabilidades, aplicando o padrão arquitetural MVC (Model-View-Controller) para garantir modularidade e escalabilidade.

## 2.1 Padrões de Design Utilizados

|  |  |
| --- | --- |
| Padrão | Descrição |
| Encapsulamento | Cada classe encapsula seus dados e responsabilidades, como Indicador e Previsor. |
| Responsabilidade Única (SRP) | Cada módulo possui uma responsabilidade clara (dados, predição, visualização). |
| Abstração | A coleta de dados da API do BACEN está abstraída na classe Indicador. |
| Separação de Camadas | A interface gráfica é separada da lógica de negócios e de predição. |

## Componentes do Sistema

### Módulos Principais

- Indicador: representa um indicador econômico, como câmbio, IPCA, IGP-M etc., e obtém os dados da API.

- PainelIndicadores: gerencia múltiplos indicadores e consolida os dados em um único DataFrame.

- PrevisorEconômico: aplica modelos estatísticos (como regressão linear) para prever valores futuros.

- AppStreamlit: exibe a interface gráfica usando Streamlit, com gráficos e filtros interativos.

## Diagrama UML – Diagrama de Classes

### Classes Representadas:

1. **Indicador**
   * Responsável por representar indicadores econômicos (como câmbio, IPCA).
   * Deve conter métodos para buscar dados via API do Banco Central (BACEN).
2. **PainelIndicadores**
   * Gerencia múltiplos objetos da classe Indicador.
   * Consolida os dados desses indicadores em um único DataFrame.
   * Provavelmente possui composição com a classe Indicador.
3. **PrevisorEconomico**
   * Utiliza dados dos indicadores para realizar predições.
   * Pode utilizar modelos estatísticos (como regressão linear).
   * Provavelmente interage com os dados consolidados de PainelIndicadores.
4. **AppStreamlit**
   * Responsável pela interface gráfica.
   * Usa Streamlit para exibir gráficos, filtros e resultados das predições.
   * Deve acessar os dados processados pelas outras classes para exibição.

## Considerações Finais

A aplicação do design de software neste projeto contribuiu para a manutenibilidade e expansão futura da solução, permitindo a adição de novos indicadores, métodos de predição mais avançados e integrações com sistemas de recomendação e alertas. O diagrama UML facilita o entendimento da arquitetura e serve como base sólida para o desenvolvimento contínuo.