

Documento de Testes e Práticas de Qualidade de Software

Projeto Final – Engenharia de Software e Arquitetura de Sistemas

Nome do Projeto: EconoVision

1. Introdução

Este documento apresenta de forma objetiva os testes realizados e as principais práticas de qualidade de software adotadas no desenvolvimento do sistema de dashboard interativo baseado na API do Banco Central. O foco principal foi garantir que os dados fossem carregados corretamente, que os gráficos fossem exibidos sem erros e que a funcionalidade de regressão linear operasse conforme o esperado.

2. Estratégia de Testes

2.1. Testes Manuais

A maior parte dos testes foi feita manualmente, simulando diferentes interações com a interface do usuário (via Streamlit), como:

- Seleção de múltiplos indicadores
- Alteração de datas de início para filtragem dos dados
- Verificação visual dos gráficos gerados
- Execução da regressão linear com diferentes combinações de dados

Esses testes foram suficientes para identificar e corrigir problemas simples, como:

- Indicadores com dados faltantes
 - Formato de data inválido
 - Falha na exibição de gráficos quando nenhum dado estava disponível
-

2.2. Testes Simples com Código

Alguns testes básicos foram implementados diretamente no código ou em notebooks de validação para garantir que:

- A função `carregar_dados()` realmente retornava DataFrames com dados válidos
- A função `_preparar_comparacao()` não quebrava quando um ou mais indicadores estavam ausentes
- A regressão linear conseguia rodar mesmo com uma quantidade reduzida de dados

Exemplo de teste simples feito durante o desenvolvimento:

```
# Validação do carregamento  
df_selic, *_ = carregar_dados()  
assert not df_selic.empty, "Erro: SELIC não foi carregado corretamente"
```

3. Práticas de Qualidade Adotadas

3.1. Organização em Módulos

O código foi dividido em três arquivos principais:

- `PreparacaoDeDados.py`: responsável por coletar e organizar os dados da API
- `graph.py`: contém os métodos de visualização (linha, barras, dispersão, boxplot e matriz de correlação)
- `regression.py`: executa a regressão linear e exibe os resultados

Isso facilitou a manutenção e leitura do projeto, separando claramente as responsabilidades.

3.2. Limpeza e Padronização

Durante o desenvolvimento, foram seguidas algumas boas práticas:

- Uso consistente de nomes de variáveis e funções
- Comentários explicando blocos importantes
- Tratamento de exceções simples (como datas inválidas ou colunas ausentes)

3.3. Testes Visuais como Verificação

A verificação visual dos gráficos foi fundamental. Por se tratar de uma aplicação interativa, testamos os seguintes pontos diretamente na interface:

- Gráficos estavam aparecendo corretamente com os dados selecionados
 - Layout do Streamlit estava funcional em diferentes tamanhos de tela
 - A regressão linear exibia uma linha ajustada com os coeficientes e métricas esperadas
-

4. Resultados Obtidos

- Todos os indicadores definidos no projeto foram testados com sucesso no dashboard.
- A regressão linear foi aplicada com sucesso em várias combinações (ex: SELIC vs IPCA, IPCA vs IGPM).

- A filtragem por data funcionou corretamente, inclusive em períodos com menos dados disponíveis.
 - Pequenos bugs como datas com NaT, gráficos em branco e valores nulos foram identificados e tratados.
-

5. Melhorias Futuras

Caso o projeto continue a ser desenvolvido, algumas práticas e testes mais robustos podem ser implementados:

- Testes unitários com pytest para cada módulo
 - Testes automatizados da interface com ferramentas como streamlit-testing
 - Linting e verificação automática com flake8 ou black
 - Monitoramento contínuo da qualidade do código com GitHub Actions
-

6. Conclusão

Apesar de o projeto não ter aplicado testes automatizados complexos, foram realizadas validações práticas e suficientes para garantir o bom funcionamento do sistema. A estrutura modular do código e a clareza da interface facilitaram a verificação manual, o que resultou em um produto final funcional, estável e adequado aos objetivos propostos.

Links e Referências

- Repositório GitHub: <https://github.com/2025-1-NCC4/Projeto1>
- API Banco Central: <https://www3.bcb.gov.br/sgspub>