

1. Objetivo

Este documento tem como objetivo apresentar os testes realizados e as práticas de qualidade de software aplicadas ao projeto interdisciplinar que analisa a correlação entre salário mínimo, consumo das famílias e endividamento no Brasil.

Com base em dados oficiais do BACEN, os testes e análises estatísticas buscam validar o funcionamento, precisão e robustez do sistema desenvolvido com arquitetura modular (MVC) métodos de práticas ágeis (Scrum).

2. Plano de Testes

O plano de testes foi estruturado para ser feito de 3 maneiras:

- testes unitários
- testes de integração
- testes de regressão.

Os testes foram feitos nas etapas de coleta, análise e visualização dos dados. Também usamos métodos estatísticos, como testes T e regressões lineares, para verificar os resultados, como aprendizado na matéria de análise inferencial de dados para analisar os dados.

3. Tipos de Testes Realizados

- Testes Unitários: Validação de funções como buscar dados ao longo do tempo, preencher valores faltantes e fazer testes com os dados.
- Testes de Integração: Validação do fluxo entre os módulos (coleta → limpeza → unificação → análise).
- Testes Estatísticos:
 - A gente queria testar se o salário mínimo médio era igual a R\$ 0 (isso é o que a hipótese inicial dizia).
 - Mas os dados mostram que a média do salário mínimo é R\$ 797,24.
 - O valor do teste (chamado de t) foi 36,205, que é bem alto — isso mostra que o resultado é muito diferente do que seria esperado se a média fosse mesmo zero.

- A análise foi feita com 246 dados, o que dá 245 graus de liberdade.
- O p-valor (que indica a chance de esse resultado ser só coincidência) foi menor que 0,00000000000000022, ou seja, praticamente zero.
- Isso significa que temos certeza estatística de que a média não é R\$ 0.
- O salário mínimo médio, com 95% de confiança, fica entre R\$ 753,86 e R\$ 840,61.

Conclusão:

A média do salário mínimo realmente é diferente — e muito maior — do que zero.

4. Práticas de Qualidade de Software Aplicadas

- Controle de documentação e teste com GitHub para rastreabilidade e colaboração.
- Utilização de ambientes isolados (Google Colab, drive e RStudio) para testes controlados.
- Aplicação de boas práticas como Clean Code dentro do colab muito bem comentado, modularização em MVC e documentação.
- Análises automatizadas com Pandas e regressões no R para validação estatística, juntamente com acompanhamento dos professores.
- Padronização e tratamento dos dados (ex: datas, valores ausentes e interpolação linear).

5. Resultados Obtidos

Os testes realizados confirmaram a robustez e confiabilidade do sistema. A análise inferencial demonstrou forte correlação entre os dados estudados. A regressão indicou que o modelo explica cerca de 80% da variação do endividamento com base em IPCA e Salário Mínimo. Os testes **T** confirmaram que os indicadores possuem médias significativamente diferentes de zero.

6. Conclusão e Recomendações

As práticas de testes e qualidade aplicadas garantiram integridade nos resultados do projeto, juntamente com um andamento de todo processo de uma maneira coerente e assertiva com o que era esperado. Para versões futuras, recomenda-se a introdução de

testes automatizados contínuos, testes de performance e usabilidade, além da inclusão de novas variáveis como taxa de juros e desemprego para refinar os modelos estatísticos.

7. Estratégia e Ferramentas de Testes Utilizadas

A estratégia de testes foi delineada com foco em garantir a qualidade do sistema e cobertura dos módulos críticos. As principais ações incluíram a definição dos tipos de testes se escolha de ferramentas adequadas para cada fase do desenvolvimento.

As camadas priorizadas para testes foram:

- **Serviços e APIs (ex: coleta de dados do BACEN via `bcb.sgs`)**
- **Regras de negócio de análise estatística (regressões, cálculos de variações)**
- **Validação e limpeza dos dados feitas no R e no COLAB (interpolação, remoção de nulos)**
- **Interface de visualização (ex: gráficos gerados com matplotlib dentro do COLAB)**

8. Ferramentas e Frameworks Utilizados

As seguintes ferramentas foram empregadas ao longo do projeto:

- **PyTest:** para testes unitários das funções em Python
- **Google Colab:** para prototipação e execução dos notebooks de análise
- **GitHub:** para versionamento e integração contínua das entregas

9. Monitoramento e Qualidade do Código

Embora o tempo de projeto tenha sido “curto” ainda foi feito da maneira mais profissional que conseguimos realizando em cada etapa da engenharia de software, foram aplicadas boas práticas e validações manuais na análise da qualidade do código. Propõe-se a futura integração com novas tecnologias que facilitem a análise das seguintes métricas:

- **Complexidade ciclomática**
- **Duplicação de código**