

# Planejamento de teste - API Calc. Imposto

## 1. Apresentação:

- Este documento descreve as estratégias, abordagens e possíveis melhorias para a API Calc. Imposto. Para fins educacionais.
- 

## 2. Objetivo:

- Correção das regras fiscais aplicadas.
  - Consistência dos valores calculados.
  - Confiabilidade da conversão de moedas.
  - Garantindo assim a progressão para novas features.
- 

## 3. Escopo de testes:

### 3.1 Dentro do escopo de testes:

- Cálculo do valor total do produto.
- Conversão de moeda estrangeira para BRL.
- Aplicação do Imposto de Importação (60%) quando valor  $\geq 250$ .
- Aplicação do ICMS (18%, ICMS travado nesta porcentagem representando ICM de SP do ano de 2024).
- Tratamento de moedas válidas.
- Isolamento da API de câmbio via mock.

### 3.2 Fora do escopo de testes:

- Interface gráfica.
  - Persistência em banco de dados.
  - Autenticação / autorização.
  - Testes de performance e carga.
  - Validação legal por estado (ICMS regional).
-

## 4. Estratégia de testes:

A estratégia será baseada em Testes Automatizados, com foco em testes unitários e isolamento dependências externas.

Será usado a abordagem de Teste White Box, abordagem essa sendo escolhida devido estarmos com o código em mãos, onde foi verificada a falta de um planejamento alinhado com algum QA. Assim, sem usar TDD e sem registros de testes unitários.

Tipos de teste aplicados:

- **Testes Unitários**
  - Regras de cálculo.
  - Impostos.
  - Conversão de valores.
- **Testes com Mock**
  - API externa de câmbio.
- **Testes de Validação**
  - Entradas inválidas.
  - Cenários de borda (boundary value).

Framework principal:

- `pytest`
- `unittest.mock`

---

## 5. Critérios de Teste

### 5.1 Critérios de Entrada

- Código refatorado com lógica separada de input/output.
- Funções puras para cálculo.
- Ambiente Python configurado.
- Pytest instalado.

### 5.2 Critérios de Saída

- 100% dos testes automatizados executando com sucesso.
  - Nenhuma falha crítica nos cálculos.
  - Regras fiscais aplicadas corretamente.
  - Cobertura mínima desejada: **≥ 80%**.
-

6. Cenários de Teste (Alto Nível)

ID	Cenário	Descrição
CT-01	Cálculo sem imposto	Valor convertido abaixo de 250
CT-02	Cálculo com imposto	Valor convertido igual ou acima de 250
CT-03	ICMS aplicado	ICMS calculado corretamente
CT-04	Moeda válida	Conversão com moeda existente
CT-05	Moeda inválida	Tratamento de erro
CT-06	API indisponível	Simulação de falha via mock

7. Riscos e Mitigações

Risco	Impacto	Mitigação
API de câmbio fora do ar	Alto	Uso de mock nos testes
Regra fiscal incorreta	Alto	Testes unitários detalhados
Mudança futura de taxa	Médio	Testes desacoplados
Entrada inválida	Médio	Validações e testes negativos

7. Ferramentas e Ambiente

- Linguagem: Python 3.x
- Framework de Teste: Pytest.

- Mock: unittest.mock.
  - Versionamento: Git / GitHub.
  - IDE: VS Code.
  - IA: Amazon Q.
- 

## **8. Plano de Evolução (Pós Qualidade Garantida)**

Após a estabilização da versão atual:

- Criação de novas features.
  - Exposição via API REST.
  - Testes de integração.
  - Pipeline CI/CD.
  - Testes de regressão automáticos.
- 

**Responsável: Kauã Raffaello - QA. Jr.**