

# DESENVOLVIMENTO DE FRAMEWORK DE TESTES DE INTEGRAÇÃO DE API

Amboni, J.

<sup>1</sup>Curso de Ciência da Computação/UNESC

Palavras chaves: Teste de software, Framework de teste, Testes de integração, Gerkin

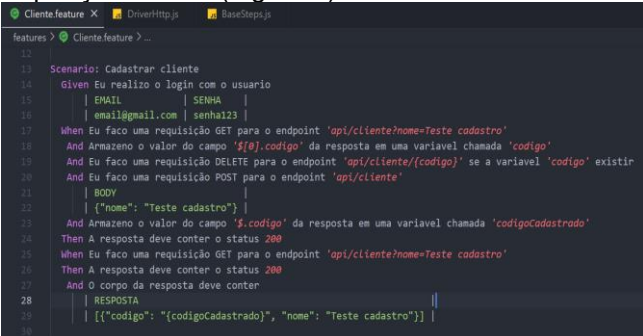
Autor: Do autor(2025)

## Introdução

No desenvolvimento de software, os erros podem aparecer em qualquer etapa do ciclo de vida do projeto. Diante disso, o teste de software se torna essencial pois oferece os métodos e ferramentas necessárias para garantir a qualidade de qualquer desenvolvimento. (TREINAWEB, [2020], [s.p.]). Estudos recentes indicam que falhas em software custam às empresas globais cerca de \$1,7 trilhão por ano. (ABC DA COMUNICAÇÃO, [2024], [s.p.]). O momento em que um bug é identificado e corrigido pode ter um impacto significativo nos custos totais do projeto. Quanto mais cedo um bug for detectado, menor será o custo para corrigi-lo. (TESTING COMPANY, [2024], [s.p.]) O projeto resultou no desenvolvimento de um framework de testes de integração que viabiliza a criação de testes automatizados por profissionais da área de qualidade sem conhecimentos especializados em automação de testes e programação, possibilitando o aumento da cobertura de testes regressivos automatizados em fases iniciais do desenvolvimento de software, resultando na identificação precoce de defeitos e na economia de tempo e recursos na correção de problemas.

## Metodologia

Por meio do framework Cucumber que possibilita a escrita de testes automatizados utilizando a sintaxe Gherkin e o ambiente de execução Node.js para a execução da linguagem de programação JavaScript, foram desenvolvidos passos em linguagem natural que realizam requisições para qualquer rota com os métodos HTTP GET, POST, PUT e DELETE e validam as respostas das requisições. Essa combinação facilitou a criação de variações de cenários de teste de integração, exigindo apenas a especificação das rotas da API e dos corpos das requisições HTTP (Figura 1).



```
1 features > Cliente.feature > ...
2
3 Scenario: Cadastrar cliente
4   Given Eu realizo o login com o usuario
5     | EMAIL | SENHA |
6     | email@gmail.com | senha123 |
7
8   When Eu faco uma requisicao GET para o endpoint 'api/cliente/nome=Teste cadastro'
9     And Armazeno o valor do campo '$[0].codigo' da resposta em uma variavel chamada 'codigo'
10    And Eu faco uma requisicao DELETE para o endpoint 'api/cliente/{codigo}' se a variavel 'codigo' existir
11    And Eu faco uma requisicao POST para o endpoint 'api/cliente'
12      | BODY |
13      | {"nome": "Teste cadastro"} |
14    And Armazeno o valor do campo '$.codigo' da resposta em uma variavel chamada 'codigoCadastrado'
15    Then A resposta deve conter o status 200
16
17   When Eu faco uma requisicao GET para o endpoint 'api/cliente/nome=Teste cadastro'
18    Then A resposta deve conter o status 200
19    And O corpo da resposta deve conter
20      | RESPOSTA |
21      | [{"codigo": "codigoCadastrado"}, {"nome": "Teste cadastro"}] |
```

Figura 1: Teste usando a sintaxe Gherkin

## Resultados e Discussão

Com o uso dos passos disponibilizados pelo framework foram criados 13 cenários de testes de integração que validam as operações de criação, leitura, alteração e exclusão de 13 rotas responsáveis pelo gerenciamento das entidades que representam os clientes, tipos de pagamento, serviços e agendamentos de uma API de agendamento de serviços. O desenvolvimento dos testes ocorreu de maneira rápida e fácil, atingindo o objetivo de possibilitar a criação de testes automatizados por qualquer pessoa da área de qualidade, o que resultará na ampliação da cobertura de testes em estágios iniciais do ciclo de desenvolvimento do software e na economia de recursos do projeto.

## Conclusão

Como resultado, o projeto buscou viabilizar uma maneira rápida e fácil de expandir a cobertura de testes em fases iniciais do ciclo de desenvolvimento de software através do uso de linguagem natural para a especificação de testes de integração automatizados sem a necessidade de especialização em automação de testes.

## Referências

- TREINAWEB. Importância dos testes de software na qualidade do sistema.** [S. l.], 2020. Disponível em: <https://encurtador.com.br/jpR8h>. Acesso em: 19 jun. 2025.
- ABC DA COMUNICAÇÃO. Startup inova na prevenção de bugs em softwares: modelo colaborativo é referência em um mercado que perde quase \$ 2 trilhões com o problema.** [S. l.], 2024. Disponível em: <https://encurtador.com.br/0QI3C>. Acesso em: 19 jun. 2025.
- TESTING COMPANY. Economize tempo e dinheiro: saiba como minimizar os gastos com a correção de bugs.** [S. l.], 2024. Disponível em: <https://encurtador.com.br/oUmyl>. Acesso em: 19 jun.

2025.