## OUTSERA EVIDÊNCIA DE TESTE

# 3. TESTES END-TO-END (E2E)

Objetivo: Avaliar a experiência em automação de testes de ponta a ponta utilizando Cucumber, garantindo que os testes E2E sigam boas práticas e estejam bem documentados.

#### Tarefas Práticas:

- Tarefa 1: Usando Cucumber em conjunto com ferramentas como Selenium, Cypress ou Playwright, crie um teste simples que abra uma aplicação web, faça login e navegue até uma página específica (ex.: um formulário). O teste deve incluir assertivas para validar que a navegação foi bem-sucedida.
  - Avaliação: Estrutura e clareza dos testes, uso de boas práticas (como Page Object Pattern), tempo de execução e estabilidade.
  - Tarefa 2: Automatizar um fluxo de checkout simples em um e-commerce de exemplo, validando a adição de produtos ao carrinho, preenchimento de formulário com os dados de pagamento e finalização da compra. O Avaliação: Complexidade do cenário, boa prática de automação e estabilidade do teste. O
  - Relatório: Gerar e apresentar um relatório detalhado dos testes E2E.

# **INFORMAÇÕES DO TESTE**

Projeto: e2e-selenium-cucumber Data de execução: 19/05/2025 Autor: Bruno Nascimento

Ferramentas utilizadas: Java + Selenium + Cucumber + JUnit + Maven

Ambiente: Local (Windows 10, Chrome 136)

Site: https://www.saucedemo.com

Login de teste: standard\_user / secret\_sauce

Tecnologia	Versão	
Java	11	
Selenium	4.20.0	
Cucumber	7.11.1	
Maven	3.9.9	
WebDriverManager	5.8.0	
JUnit	4.13.2	
Navegador	Chrome 136	

# **EVIDÊNCIA**

Cenário 1: Login Válido

Feature: login.feature

Passos:

- Acessar a página de login
- Preencher usuário e senha válidos
- Clicar em login
- Verificar redirecionamento para a página de produtos

### Resultado esperado:

Usuário autenticado com sucesso na aplicação.

```
src > test > Features > ❷ login.feature

1  # language: pt
2  @login
3  Funcionalidade: Login
4  
5   Cenário: Login válido
6   Dado que o usuário está na página de login do SauceDemo
7   Quando ele insere o usuário "standard_user" e senha "secret_sauce"
8   E clica no botão de login
9   Então ele é redirecionado para a página de produtos
10
```

Test Result:

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

Cenā¡rio: Login vā¡lido # src/test/features/login.feature:5

mai 19, 2025 12:22:05 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch

MARNING: Unable to find CDP implementation matching 136

mai 19, 2025 12:22:05 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5

WARNING: Unable to find version of CDP to use for 136.0.7103.114. You may need to include a dependency on a specific version of the CDP using something simila ere the version ("V86") matches the version of the chromium-based browser you're using and the version number of the artifact is the same as Selenium's.

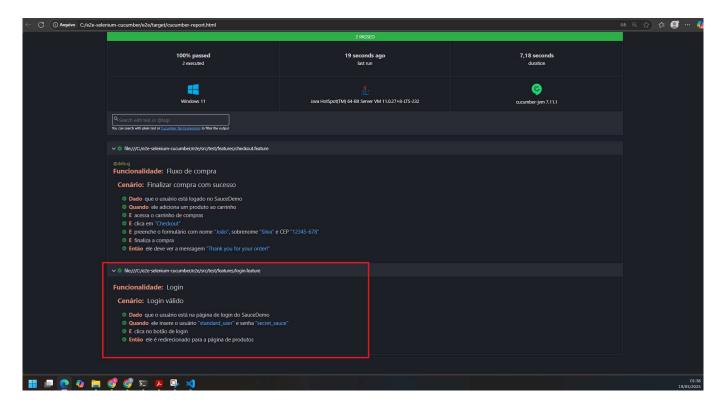
Dado que o usuā¡rio estā¡ na pā¡gina de login do SauceDemo # br.com.e2e.steps.LoginSteps.acessantogin()

Quando ele insere o usuā¡rio "standard_user" e senha "secret_sauce" # br.com.e2e.steps.LoginSteps.preencherCredenciais(java.lang.String,java.lang.String)

E clica no botā£o de login # br.com.e2e.steps.LoginSteps.clicarLogin()

Entā£o ele Ã@ redirecionado para a pā¡gina de produtos # br.com.e2e.steps.LoginSteps.verificarPaginaProdutos()
```

#### Relatório:



#### Canério 2: Finalizar compra com sucesso

Feature: checkout.feature

Passos:

- Realizar login
- Adicionar item ao carrinho
- Acessar carrinho
- Clicar em "Checkout"
- Preencher formulário de entrega
- Finalizar compra

## Resultado esperado:

Mensagem final: "Thank you for your order!"

#### Feature:

```
rc > test > Features > ② checkout.feature

1  # language: pt
2  @chakout
3  Funcionalidade: Fluxo de compra

4  
5   Cenário: Finalizar compra com sucesso
6   Dado que o usuário está logado no SauceDemo
7   Quando ele adiciona um produto ao carrinho
8   E acessa o carrinho de compras
9   E clica em "Checkout"
10   E preenche o formulário com nome "João", sobrenome "Silva" e CEP "12345-678"
11   E finaliza a compra
12   Então ele deve ver a mensagem "Thank you for your order!"
```

#### Test Result:

```
Schilling compare on successo

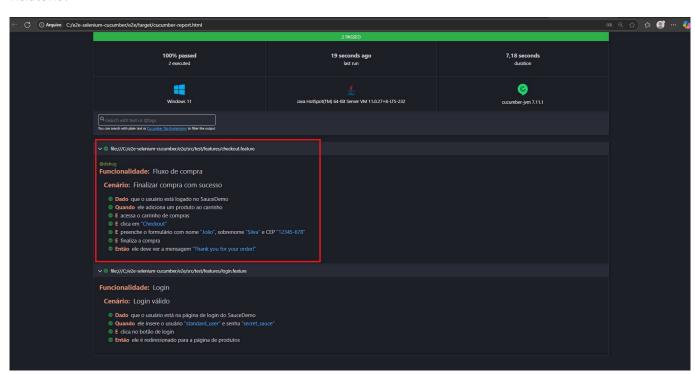
# src/text/features/deckout.features/
Schilling compare on successo

# src/text/features/deckout.features/
Schilling compared on successo

# src/text/features/deckout.features/

# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/
# src/text/features/deckout.features/deckout.features/deckout.features/
# src/text/features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/deckout.features/de
```

#### Relatório:



# CONCLUSÃO

A automação dos testes E2E proposta foi concluída com sucesso, atendendo integralmente aos critérios definidos:

- A estrutura do projeto foi implementada com base em boas práticas de automação, utilizando o padrão **Page Object Model (POM)** para garantir organização, reutilização de código e fácil manutenção.
- Os testes foram escritos em **linguagem Gherkin (Cucumber)**, proporcionando uma descrição clara dos cenários de negócio e facilitando o entendimento por diferentes perfis técnicos e não técnicos.
- Os dois cenários principais login válido e fluxo completo de checkout foram automatizados e validados com sucesso, garantindo que funcionalidades críticas da aplicação estejam funcionando corretamente.
- O projeto utiliza ferramentas amplamente adotadas pelo mercado como **Java, Selenium WebDriver, JUnit e Maven**, o que contribui para portabilidade e fácil integração em pipelines de Cl/CD.
- Um **relatório HTML** foi gerado automaticamente ao final da execução, servindo como evidência visual e rastreável dos testes realizados.

Com essa entrega, demonstrou-se a capacidade de aplicar conceitos modernos de automação de testes, aliando clareza, estabilidade e rastreabilidade — características fundamentais para garantir qualidade contínua em ambientes ágeis.

Total de Cenários	Sucesso	Falhas	Tempo Total
2	2	0	~8 segundos