

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. ○ Avaliação: Complexidade do cenário, boa prática de automação e estabilidade do teste. ○
- 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 > 🟢 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

Cenário: Login válido # src/test/features/login.feature:5
mai 19, 2025 12:22:05 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: 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 similar
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 usuário está na página de login do SauceDemo # br.com.e2e.steps.LoginSteps.acessarLogin()
Quando ele insere o usuário "standard_user" e senha "secret_sauce" # br.com.e2e.steps.LoginSteps.preencherCredenciais(java.lang.String,java.lang.String)
E clica no botão de login # br.com.e2e.steps.LoginSteps.clicarLogin()
Então ele é redirecionado para a página de produtos # br.com.e2e.steps.LoginSteps.verificarPaginaProdutos()
```

Relatório:

2 PASSED

100% passed 2 executed	19 seconds ago last run	7,18 seconds duration
Windows 11	Java HotSpot(TM) 64-Bit Server VM 11.0.27+8-LTS-232	cucumber-jvm 7.11.1

Search with text or @tags
You can search with plain text or [Cucumber tags expressions](#) to filter the output

file:///C:/e2e-selenium-cucumber/e2e/src/test/features/checkout.feature

Debug

Funcionalidade: Fluxo de compra

Cenário: Finalizar compra com sucesso

- Dado que o usuário está logado no SauceDemo
- Quando ele adiciona um produto ao carrinho
- E acessa o carrinho de compras
- E clica em "Checkout"
- E preenche o formulário com nome "João", sobrenome "Silva" e CEP "12345-678"
- E finaliza a compra
- Então ele deve ver a mensagem "Thank you for your order!"

file:///C:/e2e-selenium-cucumber/e2e/src/test/features/login.feature

Funcionalidade: Login

Cenário: Login válido

- Dado que o usuário está na página de login do SauceDemo
- Quando ele insere o usuário "standard_user" e senha "secret_sauce"
- E clica no botão de login
- Então ele é redirecionado para a página de produtos

01:38
19/05/2025

Canário 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 > 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:

```
@debug
Cenário: Finalizar compra com sucesso # src/test/features/checkout.feature:5
SLF4J(W): No SLF4J providers were found.
SLF4J(W): Defaulting to no-operation (NOP) logger implementation
SLF4J(W): See https://www.slf4j.org/codes.html#noProviders for further details.
mai 19, 2025 12:22:01 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 136
mai 19, 2025 12:22:01 AM org.openqa.selenium.chromium.ChromiumDriver lambda$new$5
WARNING: Unable to find version of CDP to use for 136.0.7103.114. You may need to include a dependency on a specific version of the CDP using something similar to `org.seleniumhq.selenium:selenium-
ere the version ("v86") matches the version of the chromium-based browser you're using and the version number of the artifact is the same as Selenium's.
mai 19, 2025 12:22:02 AM org.openqa.selenium.devtools.CdpVersionFinder findNearestMatch
WARNING: Unable to find CDP implementation matching 136
mai 19, 2025 12:22:02 AM org.openqa.selenium.chromium.ChromiumDriver lambda$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 similar to `org.seleniumhq.selenium:selenium-
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 usuário está logado no SauceDemo # br.com.e2e.steps.CheckoutSteps.usuarioLogado()
Quando ele adiciona um produto ao carrinho # br.com.e2e.steps.CheckoutSteps.adicionarProduto()
E acessa o carrinho de compras # br.com.e2e.steps.CheckoutSteps.acessarCarrinho()
E clica em "Checkout" # br.com.e2e.steps.CheckoutSteps.clicarCheckout(java.lang.String)
E preenche o formulário com nome "João", sobrenome "Silva" e CEP "12345-678" # br.com.e2e.steps.CheckoutSteps.preencherFormulario(java.lang.String,java.lang.String,java.lang.String)
E finaliza a compra # br.com.e2e.steps.CheckoutSteps.finalizarCompra()
Então ele deve ver a mensagem "Thank you for your order!" # br.com.e2e.steps.CheckoutSteps.verificarMensagem(java.lang.String)

Share your Cucumber Report with your team at https://reports.cucumber.io
Activate publishing with one of the following:

src/test/resources/cucumber.properties: cucumber.publish.enabled=true
src/test/resources/junit-platform.properties: cucumber.publish.enabled=true
Environment variable: CUCUMBER_PUBLISH_ENABLED=true
JUnit: @CucumberOptions(publish = true)

More information at https://cucumber.io/docs/cucumber/environment-variables/
Disable this message with one of the following:

src/test/resources/cucumber.properties: cucumber.publish.quiet=true
src/test/resources/junit-platform.properties: cucumber.publish.quiet=true

[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 8.274 s -- in br.com.e2e.runner.TestRunner
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- jar:3.4.1:jar (default-jar) @ e2e-selenium-cucumber ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] Building jar: C:\e2e-selenium-cucumber\e2e\target\e2e-selenium-cucumber-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- install:3.1.2:install (default-install) @ e2e-selenium-cucumber ---
[INFO] Installing C:\e2e-selenium-cucumber\e2e\pom.xml to C:\Users\Dell\m2\repository\br\com\e2e\e2e-selenium-cucumber\1.0-SNAPSHOT\e2e-selenium-cucumber-1.0-SNAPSHOT.pom
[INFO] Installing C:\e2e-selenium-cucumber\e2e\target\e2e-selenium-cucumber-1.0-SNAPSHOT.jar to C:\Users\Dell\m2\repository\br\com\e2e\e2e-selenium-cucumber\1.0-SNAPSHOT\e2e-selenium-cucumber-1.0-SNAPSHOT.jar
[INFO]
[INFO] BUILD SUCCESS
[INFO]
[INFO] Total time: 13.419 s
[INFO] Finished at: 2025-05-19T00:22:08-03:00
[INFO]
PS C:\e2e-selenium-cucumber\e2e> target/cucumber-report.html
```

Relatório:

2 PASSED

100% passed
2 executed

19 seconds ago
last run

7,18 seconds
duration

Windows 11

Java HotSpot(TM) 64-Bit Server VM 11.0.27+8-LTS-232

cucumber-jvm 7.11.1

Search with text or @tags
You can search with plain text or Cucumber Tag Expressions to filter the output

file:///C:/e2e-selenium-cucumber/e2e/src/test/features/checkout.feature

@debug

Funcionalidade: Fluxo de compra

Cenário: Finalizar compra com sucesso

- Dado que o usuário está logado no SauceDemo
- Quando ele adiciona um produto ao carrinho
- E acessa o carrinho de compras
- E clica em "Checkout"
- E preenche o formulário com nome "João", sobrenome "Silva" e CEP "12345-678"
- E finaliza a compra
- Então ele deve ver a mensagem "Thank you for your order!"

file:///C:/e2e-selenium-cucumber/e2e/src/test/features/login.feature

Funcionalidade: Login

Cenário: Login válido

- Dado que o usuário está na página de login do SauceDemo
- Quando ele insere o usuário "standard_user" e senha "secret_sauce"
- E clica no botão de login
- Então ele é redirecionado para a página de produtos

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 CI/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