

Plano de Teste- Projeto de automação

Miguel Fernandes Rita

1 Introdução

Este documento descreve os requisitos a testar, os tipos de testes definidos para cada iteração, os recursos de hardware e software a serem empregados e o cronograma dos testes ao longo do projeto. O que vai ser testado é a app “Sauce Labs Sample App” versão para Android, o site “<https://www.saucedemo.com>” e a API “<https://jsonplaceholder.typicode.com/>”.

2 Requisitos a Testar

Casos de uso do Mobile:

Identificador do caso de uso	Nome do caso de uso
id1	Login na aplicação
id2	Compras na aplicação

Casos de uso Web:

Identificador do caso de uso	Nome do caso de uso
id3	Login no site
id4	Compras no site

Casos de uso API:

Identificador do caso de uso	Nome do caso de uso
id5	Teste do method GET
id6	Teste do method POST
id7	Teste do method Put
id8	Teste do method DELETE

2.1 Use case id1 (Login na aplicação)

2.1.1 - Login com credenciais válidas

Objetivo	Fazer login com as credenciais válidas
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.1.2 - Login com credenciais invalidas e validação da mensagem de erro

Objetivo	Fazer login com credenciais invalidas e validação de mensagem de erro
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

```

Run MyRunnerTest
com.qa.runners.MyRunnerTest 4 min 11 sec 3 tests passed 3 tests total, 4 min 11 sec
  Login scenarios 4 min 11 sec
    Login with invalid user name 1 min 18 sec
    Login with invalid password 1 min 2 sec
    Login with valid username and password 1 min 50 sec

[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Got response with status 200: {"sessionId":"cb91
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Got response with status 200: {"sessionId":"cb91
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Got response with status 200: {"sessionId":"cb91
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Screen already unlocked, doing nothing
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Starting 'com.swaglabsmobileapp/com.swaglabsmob
[6823284d][ADB] Getting IDs of all 'com.swaglabsmobileapp' processes
[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] 'com.swaglabsmobileapp' is already running and n
[6823284d][AppiumDriver@97b7f] New AndroidUiautomator2Driver session created successfully, session 6823
[6823284d][AppiumDriver@97b7f] Event 'newSessionStarted' logged at 1747407291343 (15:54:51 GMT+0100 (We
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Cached the protocol value 'W3C' for the new sess
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Responding to client with driver.createSession()
[6823284d][HTTP] <- POST /session/200 3974 ms - 1264
[6823284d][HTTP] --> POST /session/6823284d-f2f5-4aec-95d3-3c342b3b5f40/appium/start_recording_screen
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Calling AppiumDriver.startRecordingScreen() with
[6823284d][ADB] Getting IDs of all 'screenrecord' processes
[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Screen recording has not been previously started
[6823284d][ADB] Getting IDs of all 'screenrecord' processes
[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d][ADB] Building screenrecord process with the command line: adb -P 5037 -s emulator-5554 shel
[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d][ADB] Running "/Users/miguel.fernandes/Library/Android/sdk/platform-tools/adb -P 5037 -s emu"
[6823284d] [38;5;-137n[AndroidUiautomator2Driver@9b5d] Responding to client with driver.startRecording
[6823284d][HTTP] <- POST /session/6823284d-f2f5-4aec-95d3-3c342b3b5f40/appium/start_recording_screen

```

2.2 Use Case id2 (Compras na aplicação)

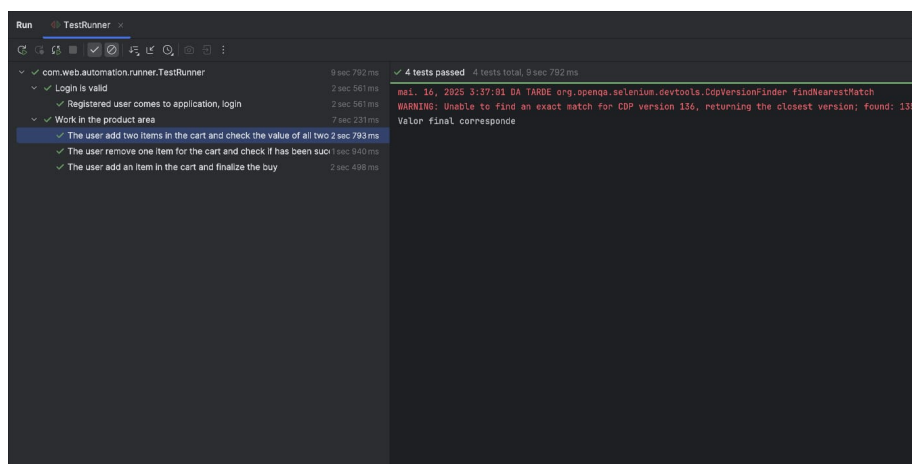
2.2.1 - Adicionar produtos ao carrinho e verificar se está correto

Objetivo	Adicionar produtos ao carrinho e verificar se está correto
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta

Objetivo	Adicionar produtos ao carrinho e verificar se está correto
Responsável	Miguel Rita

2.2.2 - Finalizar uma compra até ao passo de review

Objetivo	Finalizar uma compra até ao passo de review
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita



2.3 Use Case id3 (Login no site)

2.3.1 - Login com usuário padrão e senha correta

Objetivo	Login com usuário padrão e senha correta
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.4 Use Case id4 (Compras no site)

2.4.1 - Adicionar dois itens ao carrinho e validar o total

Objetivo	Adicionar dois itens ao carrinho e validar o total
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.4.2 - Remover um item do carrinho e validar atualização

Objetivo	Remover um item do carrinho e validar atualização
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.4.3 - Finalizar pedido até à tela de confirmação

Objetivo	Finalizar pedido até à tela de confirmação
Técnica	Automática
Estágio do teste	Sistema
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

The screenshot shows an IDE window titled 'Run' with a sub-window 'MyRunnerTest'. The left pane shows a test suite 'com.qa.runners.MyRunnerTest' with two tests: 'Product scenarios' (2 min 0 sec) and 'Add Product in the Cart and Validate' (1 min 6 sec). The right pane shows the test results and logs. The logs indicate that the test 'Add Product in the Cart and Validate' passed successfully. The logs also show the test setup, including the creation of an 'AppiumDriver' and the execution of the 'GET /status' endpoint.

```

@test
Scenario Outline: Add Product in the Cart and Validate
Starting Server
Capabilities (app: /Users/miguel.fernandes/Doc..., appium:appActivity: com.swaglabsmobileapp.Splas...
[HTTP] --> GET /status {}
[AppiumDriver@9564e] Calling AppiumDriver.getStatus() with args: {}
[AppiumDriver@9564e] Responding to client with driver.getStatus() result: {"ready":true,"message":"The
[HTTP] <-- GET /status 200 0 ms - 200
[HTTP] Request idempotency key: efe77d43-4aaa-4abe-b567-36c7e8d23771
[HTTP] --> POST /session {"capabilities":{"firstMatch":[{"}],"alwaysMatch":{"platformName":"ANDROIDID","p
[AppiumDriver@9564e] Calling AppiumDriver.createSession() with args: [null,null,{"firstMatch":[{"}],"alw
[AppiumDriver@9564e] Event 'newSessionRequested' Logged at 1747486684392 (15:44:44 GMT+0100 (Western Eu
[Appium] Attempting to find matching driver for automationName 'UIAutomator2' and platformName 'ANDROI
[Appium] The 'uiautomator2' driver was installed and matched caps.
[Appium] Will require it at /Users/miguel.fernandes/.appium/node_modules/appium-uiautomator2-driver
[Appium] Requiring driver at /Users/miguel.fernandes/.appium/node_modules/appium-uiautomator2-driver/b
[AppiumDriver@9564e] Appium v2.17.1 creating new AndroidUiautomator2Driver (v4.8.2) session
[AppiumDriver@9564e] Checking BaseDriver versions for Appium and AndroidUiautomator2Driver
[AppiumDriver@9564e] Appium's BaseDriver version is 9.16.4
[AppiumDriver@9564e] AndroidUiautomator2Driver's BaseDriver version is 9.16.4
[38;5;~139m[AndroidUiautomator2Driver@95629]
[38;5;~139m[AndroidUiautomator2Driver@95629] Creating session with W3C capabilities: {
  "alwaysMatch": {
    "platformName": "ANDROIDID",
    "appium:appPackage": "com.swaglabsmobileapp",
    "appium:fullReset": false,
    "appium:automationName": "UIAutomator2"
  }
}

```

2.5 Use Case id5 (Teste do method GET)

2.5.1 - GET /post e validar que retorna 100 posts

Objetivo	GET /post e validar que retorna 100 posts
Técnica	Automática
Estágio do teste	Integração
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.5.2 - GET /posts/1 e validar o titulo

Objetivo	GET /posts/1 e validar o titulo
Técnica	Automática
Estágio do teste	Integração
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.6 Use Case id6 (Teste do method POST)

2.6.1 - POST /posts criando um novo post e validar o retorno do id

Objetivo	POST /posts criando um novo post e validar o retorno do id
Técnica	Automática
Estágio do teste	Integração
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.7 Use Case id7 (Teste do method PUT)

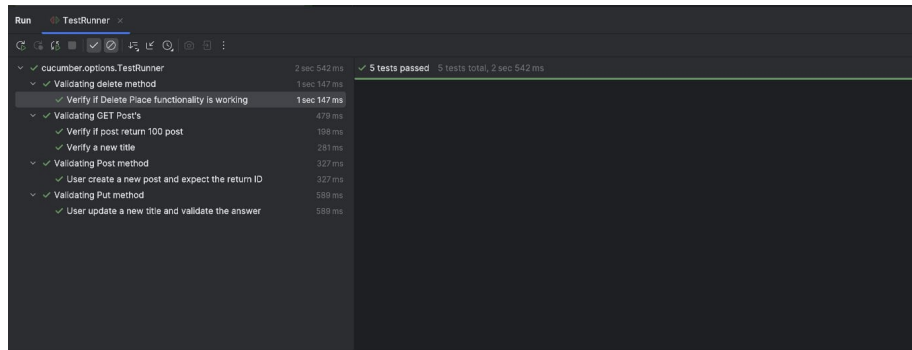
2.4.1 - PUT /posts/1 atualizando o conteudo e validando a resposta

Objetivo	PUT /posts/1 atualizando o conteudo e validando a resposta
Técnica	Automática
Estágio do teste	Integração
Abordagem do teste	Caixa preta
Responsável	Miguel Rita

2.8 Use Case id4 (Compras no site)

2.8.1 - DELETE /posts/1 e validar o status code

Objetivo	DELETE /posts/1 e validar o status code
Técnica	Automática
Estágio do teste	Integração
Abordagem do teste	Caixa preta
Responsável	Miguel Rita



3 Tipos de teste

3.1- Testes funcionais

- Use case id1 (Login na aplicação);
- Use case id2 (Compras na aplicação);
- Use case id3 (Login no site);
- Use case id4 (Compras no site);

3.2- Testes de integração

- Use case id5 (Teste do method GET);
- Use case id6 (Teste do method POST);
- Use case id7 (Teste do method PUT);
- Use case id8 (Teste do method DELETE);

4 Recursos

4.1 - Ambiente de teste - Software e Hardware

Os testes foram executados num ambiente de desenvolvimento e validação com as seguintes características:

Hardware: - Equipamento: MacbBook Pro(2021) - Processador: Chip Apple M1 Pro - Memória RAM: 16GB - Armazenamento: 512GB HD

Sistema Operativo: - macOS Sequoia Versão 15.4.1

Browsers utilizados para o testes web: - Google Chrome versão 136.0.7103.114

Emulador para os testes Android: - Android Emulator via Android Studio -
Dispositivo Virtual: Pixel9 - Sistema Android: 15.0

4.2 - Ferramenta de teste

As seguintes ferramentas foram utilizadas para a execução dos testes: - IDE: IntelliJ IDEA - para desenvolvimento e execução dos testes automatizados; - Framework: BDD com Java e Cucumber - escrita e automatação de testes baseados em cenários Gherkin

Testes Mobile para Android: - Appium Inspector: utilizado para a inspeção da interface e seus elementos da aplicação mobile em teste. - Android Emulator: execução dos testes num ambiente simulado

Teste Web: - Google Chrome: browser principal para os testes automatizados; Testes de API: - Postman: para execução validação e verificação de chamadas a serviços RESTfull e consequente análise de respostas

5 - Cronograma

Tipo de teste	data de início	data de fim	Responsável
Planear os testes	24/04/2025	24/04/2025	Miguel Rita
Escrita dos use cases	02/05/2025	08/05/2025	Miguel Rita
Implementação dos testes	08/05/2025	15/05/2025	Miguel Rita
Execução dos testes	08/05/2025	15/05/2025	Miguel Rita
avaliação dos testes	15/05/2025	16/05/2025	Miguel Rita