

# O TESTADOR DE HOGWARTS

*A MAGIA DA AUTOMATIZAÇÃO*



**RICHARD BARBOSA**

# Guia de Automação para QA Testers



Se você é um QA Tester em busca de aprimorar suas habilidades de automação de testes, veio ao lugar certo! Neste guia, vamos explorar de maneira rápida as principais práticas de automação com exemplos de código para ajudá-lo a automatizar seus testes de forma eficiente.



# Introdução à Automatização de Testes



# Primeiros Passos com Selenium WebDriver em Python

Automatizar testes é essencial para garantir a qualidade do software de forma eficiente. Vamos começar com um exemplo simples de automatização de teste usando Selenium WebDriver em Python.

```
● ● ● automatizacao.py

from selenium import webdriver

# Inicializar o navegador
driver = webdriver.Chrome()

# Abrir uma página da web
driver.get("https://www.exemplo.com")

# Encontrar um elemento e interagir com ele
search_box = driver.find_element_by_name("q")
search_box.send_keys("Exemplo de busca")
search_box.submit()

# Verificar se o resultado esperado é exibido
assert "Resultado da busca" in driver.title

# Fechar o navegador
driver.quit()
```



# Testes de Interface do Usuário (UI)



2



# Garantindo a Qualidade da Aplicação para o Usuário Final

A automatização de testes de interface do usuário é crucial para garantir que a aplicação funcione corretamente para o usuário final. Vamos ver um exemplo de teste de interface do usuário usando Selenium WebDriver.

● ● ● automatizacao.py

```
from selenium import webdriver

# Inicializar o navegador
driver = webdriver.Chrome()

# Abrir uma página da web
driver.get("https://www.exemplo.com")

# Encontrar um elemento e verificar se está visível
assert driver.find_element_by_id("elemento").is_displayed()

# Fechar o navegador
driver.quit()
```



# Testes de API



# Verificando Chamadas e Resultados

Testes de API verificam se as chamadas de API estão retornando os resultados esperados. Aqui está um exemplo simples usando a biblioteca requests em Python.

● ● ● automatizacao.py

```
import requests

# Fazer uma chamada GET para uma API
response = requests.get("https://api.exemplo.com/data")

# Verificar o código de status da resposta
assert response.status_code == 200

# Verificar o conteúdo da resposta
assert "Dados esperados" in response.text
```





# Testes de Banco de Dados



# Garantindo a Integridade e Funcionalidade

Os testes de banco de dados garantem que as operações de banco de dados estejam funcionando conforme o esperado. Veja um exemplo usando a biblioteca sqlite3 em Python.

```
● ● ● automatizacao.py

import sqlite3

# Conectar ao banco de dados
conn = sqlite3.connect('exemplo.db')
cursor = conn.cursor()

# Executar uma consulta SQL
cursor.execute("SELECT * FROM tabela")

# Verificar se os resultados estão corretos
assert len(cursor.fetchall()) == 10

# Fechar a conexão com o banco de dados
conn.close()
```



# Agradecimientos



# OBRIGADO POR LER ATÉ AQUI



Esse Ebook foi gerado por IA, e diagramado por humano.

O passo a passo se encontra no meu Github.  
Esse conteúdo foi gerado com fins didáticos de construção, não foi realizado uma validação cuidadosa humana no conteúdo e pode conter erros gerados por uma IA.



<https://github.com/RichardMSBarbosa/prompts-recipe-to-create-a-ebook>

