

Exercício GUI Testing - Cypress e Selenium

Introdução

O Sylius¹ é um framework de comércio eletrônico de código aberto, que fornece uma estrutura modular e flexível para construir lojas online personalizadas. O Sylius é altamente configurável e possui recursos avançados para gerenciamento de catálogo, pedidos, pagamentos e envio, entre outros aspectos do comércio eletrônico. Ele é amplamente utilizado pela comunidade de desenvolvedores para criar lojas online escaláveis e adaptáveis. **Neste exercício, vamos realizar testes de GUI numa loja recém criada pelo Sylius (mostruário), utilizando o framework Cypress e Selenium.**

O Cypress² é um framework de automação de testes de GUI de código aberto que permite a escrita e execução de scripts de teste de forma rápida e confiável. Ele se destaca por sua abordagem de execução dentro do navegador, permitindo interações em tempo real com os elementos da interface. Além disso, ele oferece uma API intuitiva para escrever testes em JavaScript e suporta várias asserções para verificar os resultados esperados.

O Selenium³ é um framework de automação de teste de código aberto amplamente utilizado para testes de GUI em aplicativos web. Ele permite a interação com os elementos da GUI, simulando ações do usuário e verificando automaticamente resultados. O Selenium possui uma variedade de componentes, sendo o Selenium WebDriver o mais utilizado. Ele suporta várias linguagens de programação e permite a automação de testes em diferentes navegadores. O Selenium é conhecido por sua compatibilidade e flexibilidade para testar aplicativos web em diferentes ambientes. No exercício vamos utilizar o Selenium na linguagem de programação JavaScript Mocha.

Aplicação alvo

Para realizar o exercício, vocês vão utilizar a loja de mostruário fornecida pelo Sylius. A loja possui duas partes: a parte de vendas e a parte administrativa. Na parte de vendas é possível realizar ações de um usuário interessado em comprar os produtos expostos, como buscar por produtos específicos, adicioná-los no carrinho de compras, fazer o check in e etc. Na parte administrativa podemos gerenciar a parte de vendas, sendo possível cadastrar e editar produtos, categorias, cupons e diversas outras funcionalidades.

¹ <https://sylius.com>, <https://github.com/Sylius/Sylius>

² <https://www.cypress.io>

³ <https://selenium.dev>

Sign up to our newsletter and grab -20% off

Login

Register



\$0.00

T-shirts ▾

Caps ▾

Dresses

Jeans ▾



Brace yourself! Winter is coming

[Check out the new collection](#)

Search

CATALOG

- Taxons
- Products
- Inventory
- Attributes
- Options
- Association types

SALES

- Orders
- Payments
- Shipments

CUSTOMER

- Customers
- Groups

MARKETING

- Cart promotions
- Catalog promotions
- Product reviews

CONFIGURATION

- Channels
- Countries
- Zones

View your store

Search products...

John ▾

Dashboard

Overview of your store

Sales summary

2 Weeks Month **Year**

\$878.49 SALES	4 PAID ORDERS	21 CUSTOMERS	\$219.62 AVERAGE ORDER VALUE
--------------------------	-------------------------	------------------------	--

Month	Sales
1.2023	580
2.2023	200
5.2023	90

Products

Orders

Customers

Cart promotions

New customers

New orders

Para facilitar a utilização da aplicação alvo, foi criada uma imagem docker capaz de subir todos os serviços necessários para aplicação (php, nginx e banco), além de instalar a última versão disponível da aplicação, tudo em um único container. O Dockerfile dessa imagem encontra-se nesse repositório⁴, juntamente com os comandos necessários para realizar o build, run e carregar os dados de exemplo no banco. É necessário ter o Docker instalado na máquina. Aconselhamos utilizar uma distribuição linux, como Ubuntu.

Aqueles que tiverem dificuldades em subir a aplicação devem procurar a orientação de Thiago (thiago.moura@copin.ufcg.edu.br).

Objetivo do exercício

Os alunos serão divididos em duplas conforme presente nesta planilha. Cada dupla será direcionada a uma seção da parte administrativa (produtos, categorias, cupons e etc) e será responsável por desenvolver, para essa seção, no **mínimo** 10 casos de teste usando o Cypress e os mesmos casos de teste usando o Selenium, totalizando no **mínimo** 20 casos de teste. Cada caso de teste deve conter no **mínimo** 4 interações com a aplicação, ou seja, quatro ações executadas no site, como clicar em botões, preencher inputs, navegar entre páginas etc. Além das interações, cada caso de teste deve incluir pelo menos um **assert**, ou seja, uma verificação para validar se o resultado esperado foi alcançado.

As duplas têm liberdade para criar testes nas duas partes da loja, por exemplo: cadastrar um novo produto na parte de administração e adicionar o produto ao carrinho na parte de vendas. O objetivo é que as duplas desenvolvam uma suíte de teste para a seção a qual eles foram direcionados, abrangendo diferentes funcionalidades e cenários de uso daquela seção. Os respectivos direcionamentos estão publicados nesta planilha.

Para executar o exercício, vocês receberão um starter kit! Trata-se de um repositório⁵ com o Cypress e Selenium configurado, scripts de teste criados para duas frameworks, para cada seção direcionada. Os scripts contêm o primeiro caso de teste que servirá como exemplo. Ou seja, não tem como tirar 0. Vocês devem completar seus respectivos scripts com no mínimo **+ 9 casos de teste** para cada framework.

⁴ <https://github.com/mourats/sylius-showcase-docker>

⁵ <https://github.com/mourats/gui-testing-cypress-selenium>

Entrega do exercício

A entrega do exercício deve ser feita neste formulário. Apenas um integrante da dupla deve preencher o formulário e enviar. O prazo para entrega é 06/06/2023 às 23:59. No formulário está descrito o que deve ser entregue, mas para deixar mais claro ainda, você deve entregar um zip do starter kit contendo a implementação dos seus casos de teste e configurações feitas. Além disso, você deve descrever as diferenças e dificuldades que vocês sentiram ao implementar os casos de teste nas duas frameworks.