

Trabalho Prático 2 | Testes de Software

Integrantes: Daniella Rodrigues de Melo, Italo Lelis e Júlia Martins Reis.

Engenharia de Software, Prof. Cleiton Tavares.

Link para o vídeo

<https://drive.google.com/file/d/1O5yygsAz8AXwqOmYwFbrl5AHly7dl0VA/view?usp=sharing>

Sistema de software web

1. Sobre o software

Projeto escolhido: <https://github.com/devfilsk/java-supermercado>

O software escolhido é um sistema de gerenciamento de Vendas de Supermercado, desenvolvido como projeto final da disciplina de Fundamentos de Orientação a Objetos dos autores Filipe Maciel e Richiely Batista. Esse sistema possui Estoque, Controle de Venda, Operadores de Caixa, Funcionários, Gerentes, Leitor, Produto sendo produto unitário e produto quilo, Caixa, Balança, entre outros.

2. Casos de teste

O projeto possui 17 classes, das quais 11 foram testadas, e 84 funções, das quais 62 foram testadas.

Foram realizados 62 casos de testes que englobam Balança, Caixa, Carrinho de Compras, Estoque de Produtos, Funcionário, Gerente, Leitor, Operador de Caixa, Produto Quilo, Supermercado e Venda.

3. Cobertura de testes

Os cálculos de cobertura foram feitos utilizando a ferramenta Coverage presente no Eclipse utilizando o framework JUnit.

Coverage atingido: 55,9%.

Sistema de software mobile

1. Sobre o software

Foi escolhido um software desenvolvido por outras pessoas e disponível no GitHub. Esse software pode ser encontrado pelo link:

<https://github.com/innFactory/react-native-quick-sample> .

O software consiste de uma simples aplicação para dispositivos móveis que adiciona cores geradas aleatoriamente a uma lista. Essa lista pode, de acordo com a necessidade do usuário, ser visualizada.

2. Casos de teste

Os casos de teste podem ser encontrados dentro da pasta `__test__`. Para a realização dos testes, foi utilizada a ferramenta **Mocha**.

Para rodar os testes automatizados, você deve rodar o comando: ***yarn test***.

3. Cobertura de testes

O cálculo da cobertura foi realizado utilizando a própria ferramenta para a escrita de testes automatizados. Atualmente, a cobertura está em **XX%**.

É possível visualizar a cobertura de testes automatizados utilizando o comando: ***yarn coverage***.

4. Dependências

A aplicação foi desenvolvida em React Native (Javascript). Você precisará do Node para rodar a aplicação e as dependências associadas ao projeto podem ser instaladas com o comando ***yarn install***.