Zé Delivery

**Estratégia de teste**

**Histórico de revisão**

|  |  |  |  |
| --- | --- | --- | --- |
| Data | Versão | Autor | Descrição |
| 15/06 | 1.0 | Jessica Alauk | Criaçao do documento |
| 16/06 | 1.1 | Jessica Alauk | Alteração do Sumário |
| 16/06 | 1.2 | Jessica Alauk | Mudança das funcionalidades no item 1 |
|  |  |  |  |

Sumário

[1. Escopo 3](#_Toc43121140)

[2. Abordagem de teste 4](#_Toc43121141)

[3. Ambiente de teste 5](#_Toc43121142)

[4. Ferramentas de teste 6](#_Toc43121143)

[5. BDDs 7](#_Toc43121144)

# Escopo

Principais cenários

* Login;
* Criação de login;
* Efetuar pedido.

Funcionalidades

Login:

* Efetuar login com sucesso;
* Efetuar login com dados inválidos;
* Efetuar login com dados em branco.

Criação do login:

* Validar criação do login com sucesso

Efetuar pedido:

* Efetuar pedido com sucesso;
* Efetuar pedido com valor mínimo estipulado;
* Não efetuar pedido.

# Abordagem de teste

Processo de teste

Os testes são feitos com o conceito “BDD”, validando o comportamento da funcionalidade como um todo.

Níveis de teste

O teste será feito nos níveis de sistema e aceitação

Tipos de testes

Serão feitos os seguintes testes:

* Funcionais – positivos e negativos;

Abordagem de teste

Focaremos a abordagem de teste no método caixa preta para testes de UI.

Automação

As funcionalidades que serão automatizadas são:

* Login, por ser um teste que não queima massa de dados.

A configuração do framework de automação encontra-se no arquivo README.md.

# Ambiente de teste

O ambiente a ser utilizado será o de produção conforme passado por negócio.

O ambiente escolhido para testes é o mobile Android, pois é o mais utilizado conforme pesquisa de mercado.

# Ferramentas de teste

Para a automação funcional será utilizado as ferramentas abaixo:

* JUnit para o main;
* Cucumber para o BDD;
* Appium para a execução da automação no mobile Android;
* Linguagem de programação Java;
* Maven para gerenciamento de bibliotecas.

Dúvidas sobre a instalação e execução, ler o documento README.md.

# BDDs

Todos os BBDs foram escritos na linguagem gherkin. Os arquivos estão no projeto do framework de automação (vide arquivo README.md)