



## PLANO DE TESTES

### Objetivo

O objetivo deste documento é apresentar um plano de testes para o Projeto nomeado Digital Booking, desenvolvido no âmbito da disciplina Projeto Integrador (PI), requisito para a conclusão do primeiro ano de curso do Certified Tech Developer da Digital House Brasil.

Este documento apresenta os objetivos abaixo:

- Identificar as funcionalidades a serem testadas no projeto;
- Listar as ferramentas e/ou tecnologias a serem utilizadas nos testes;
- Estimar a cobertura de testes necessária para mitigar defeitos na aplicação;

### Escopo

O plano de testes detalhado neste documento refere-se aos testes desenvolvidos no projeto integrador Digital Booking. A execução correta dos testes assegura a correta integração entre todas as áreas do projeto.

Quanto aos níveis, estão previstos testes unitários, de integração e end-to-end.

A cada sprint devem ser executados Testes de Fumaça (testes de requisitos).

A partir de segunda sprint, além dos testes adicionais definidos de acordo com o backlog do produto, deverão ser realizados Testes de Regressão das funcionalidades já implementadas, a fim de verificar se as alterações de código e as novas funcionalidades introduzidas não danificaram as funcionalidades anteriores.

Também estão previstos testes exploratórios manuais, que devem ser realizados de modo a testar vários cenários possíveis para cada funcionalidade, devendo incluir tanto os cenários positivos quanto os cenários negativos.

Os testes de código deverão ser realizados com JUnit, das seguintes classes e métodos:

- Categoria Service:
  - o Cadastrar categoria;
  - o Listar todas as categorias;
  - o Buscar categoria por id;
- Cidade Service:

- o Cadastrar cidade;
  - o Listar todas as cidades;
  - o Buscar cidade por nome;
- Produto Service;
  - o Cadastrar produto;
  - o Listar todos os produtos;
  - o Buscar produtos por id;
  - o Buscar produtos por id da categoria;
  - o Buscar produtos por id da cidade;
  - o Buscar produtos de acordo com as datas inicial e final;
  - o Buscar produto por id da cidade e de acordo com as datas inicial e final;
  - o Buscar produtos por id da categoria e de acordo com as datas inicial e final;
- Reserva Service;
  - o Cadastrar reserva;
  - o Buscar reserva por id;
- Usuário Service:
  - o Cadastrar usuário;
  - o Buscar usuário por id.

As API's desenvolvidas para a aplicação deverão ter seus endpoints testados utilizando o Postman e documentados através do Swagger. São eles:

- API de categoria:
  - o Adicionar categoria;
  - o Listar todas as categorias;
  - o Editar categoria;
  - o Eliminar categoria;
- API de Cidade:
  - o Adicionar cidade;
  - o Listar todas as cidades;
- API de Produto;
  - o Adicionar produto;
  - o Listar todos os produtos;
  - o Buscar produtos por nome da categoria;

- o Buscar produtos por id da categoria;
  - o Buscar produtos por nome da cidade;
  - o Buscar produtos por id da cidade;
  - o Buscar produtos por id da cidade e id da categoria;
  - o Buscar produto por id da cidade e de acordo com as datas inicial e final;
- API de Reserva;
  - o Cadastrar reserva;
  - o Buscar reserva por id do produto;
- API de Usuário:
  - o Cadastrar usuário;
  - o Login de usuário.

As telas da interface de usuário a serem testadas são as seguintes:

- Tela de registro do usuário;
- Tela de login do usuário;
- Tela de Home;
- Tela de filtro dos produtos;
- Tela de detalhes dos produtos;
- Tela de cadastro de reserva de um produto;
- Tela de minhas reservas;
- Tela de cadastro de um produto (apenas usuário ADMIN).

Deve-se realizar também o teste de responsividade das UIs para dispositivos móveis (tablet, celular), conforme os requisitos do projeto.

Os testes automatizados E2E devem ser realizados utilizando as ferramentas Selenium WebDriver e/ou Robot Framework.

## **Ferramentas**

- JUnit
- Postman
- Swagger
- Selenium WebDriver
- Robot Framework

## **Cobertura**

Para mensurar a qualidade da aplicação recomenda-se uma cobertura de código de no mínimo 40%. Objetiva-se atingir a maior cobertura possível de testes unitários, sendo esta a base da pirâmide de testes, de acordo com as boas práticas de testes de software. Entre testes caixa preta e caixa branca espera-se atingir uma cobertura de testes média de 85%.