

---

# Plano de Testes - PT

## IDENTIFICAÇÃO DO PROJETO

### Projeto

- Máquina de Café - Devine Café

### Disciplina do Projeto

- Métodos Formais de Engenharia de Software

### Equipe do Projeto

- ALLYSON BRUNO DE FREITAS FERNANDES - 2024012632;
- ANDREY DE OLIVEIRA SABINO - 2020010859;
- GEÍSA MORAIS GABRIEL - 2024012594;
- KLEBSON DAVI DE SOUZA MAGALHÃES - 2022011458;
- LÍVIA BEATRIZ MAIA DE LIMA - 2024012596;
- PEDRO DAMIÃO DE OLIVEIRA LUZ - 2021022519.

## HISTÓRICO DE REGISTROS

Versão	Data	Autor	Descrição
{1.0}	09/05/2025	Lívia Maia	Preenchimento do documento

---

## SUMÁRIO

<b>1. PLANO DE TESTES DO DEVINE CAFÉ.....</b>	<b>3</b>
<b>2. RESUMO.....</b>	<b>3</b>
<b>3. PESSOAS ENVOLVIDAS.....</b>	<b>3</b>
<b>4. AMBIENTE E FERRAMENTAS.....</b>	<b>3</b>
4.1. Ambiente.....	3
4.2. Ferramentas.....	3
<b>5. RECURSOS NECESSÁRIOS.....</b>	<b>4</b>
<b>6. CRITÉRIOS USADOS.....</b>	<b>4</b>
<b>7. RISCOS.....</b>	<b>5</b>
<b>8. METODOLOGIA DE TESTES.....</b>	<b>5</b>
<b>9. ABORDAGEM DOS TESTES.....</b>	<b>5</b>
<b>10. RESULTADOS DOS TESTES.....</b>	<b>5</b>

## 1. PLANO DE TESTES DO DEVINE CAFÉ

Este documento tem como objetivo descrever as atividades e planejamento para a execução dos testes no Devine Café - Sistema de Gerenciamento de uma cafeteria.

## 2. RESUMO

As atividades de Verificação e Validação de software têm como objetivo garantir a qualidade do produto por meio da identificação de erros no sistema, possibilitando que os desenvolvedores realizem as correções necessárias. Dentre essas atividades, destaca-se o teste de software como um dos principais meios para detectar falhas. No sistema do Devine Café, será elaborado e executado um plano de testes com base nos requisitos funcionais e não funcionais definidos. A documentação resultante abrangerá a organização do processo, a metodologia adotada, as responsabilidades envolvidas e os resultados obtidos conforme o planejado.

## 3. PESSOAS ENVOLVIDAS

Os testadores em questão foram selecionados por possuírem conhecimento avançado sobre os procedimentos dos testes e capacidade para executá-los de forma adequada.

- ALLYSON BRUNO DE FREITAS FERNANDES
- LIVIA BEATRIZ MAIA DE LIMA

## 4. AMBIENTE E FERRAMENTAS

A execução dos testes contará com o apoio de ferramentas compatíveis com o ambiente de desenvolvimento e a linguagem de programação utilizada. Os subtópicos a seguir detalham as tecnologias envolvidas e o ambiente onde os testes serão realizados.

### 4.1. Ambiente

Máquina	Processador	Armazenamento	Sistema Operacional
Acer Nitro 5	AMD Ryzen 7 4800H	1 TB de HD e 256 SSD e 16 GB de RAM	Windows 10
Acer Aspire 5	Intel Core i5	480 SSD e 8 GB de RAM	Windows 10

### 4.2. Ferramentas

Ferramenta	Descrição
xUnit	Framework para realização de testes unitários.
Selenium	Framework portátil para testar aplicativos web.
Lighthouse	Ferramenta automatizada de código aberto para medir a qualidade de páginas da web.
DevTools	Ferramenta que inspeciona, ajusta e altera os estilos dos elementos na página Web.
SonarQube	Ferramenta de análise estática do código

## 5. RECURSOS NECESSÁRIOS

Para a efetivação dos testes, é fundamental que determinados recursos estejam previamente definidos e disponíveis. Esses recursos foram organizados conforme descrito a seguir:

- **Recursos Humanos (Testadores):** Os testes deverão ser conduzidos por pessoas que possuam, no mínimo, familiaridade básica com a aplicação, bem como capacidade para interpretar o funcionamento dos testes.
- **Equipamentos (Máquinas):** É necessário disponibilizar, no mínimo, uma máquina por testador.
- **Configuração do Ambiente:** Cada máquina deve estar devidamente configurada com o ambiente necessário para a realização dos testes, conforme os requisitos do sistema.
- **Documentação do Sistema:** Os testadores devem ter acesso total à documentação oficial do sistema, a fim de garantir embasamento técnico.
- **Ferramentas de Teste:** As ferramentas utilizadas deverão ser selecionadas de acordo com os objetivos e tipos de teste planejados.

## 6. CRITÉRIOS USADOS

Para a modelagem dos testes, foram estabelecidos os seguintes critérios:

- **Quantidade total de testes:** Com base nas diretrizes descritas na documentação do sistema DEVINE CAFÉ, prevê-se a realização de, no mínimo, 100 testes no total. Cada testador deverá, portanto, planejar e executar pelo menos 50 testes.
- **Quantidade mínima de casos de testes a serem realizados:** Considerando o conjunto de funcionalidades do sistema e contemplando tanto cenários válidos quanto inválidos, estima-se a necessidade de pelo menos 20 casos de teste distintos.

- 
- **Critérios quantitativos de teste:** A avaliação dos testes estruturais será baseada na cobertura de código, com a meta de alcançar no mínimo 70% de cobertura total de linhas executadas.
  - **Critérios qualitativos de teste:** A partir da utilização de ferramentas de análise estática, espera-se identificar, no mínimo, 10 ocorrências relacionadas à qualidade do código-fonte, visando aprimorar aspectos como legibilidade, manutenibilidade e conformidade com boas práticas.

## 7. RISCOS

Possíveis contingências para a efetivação do plano de teste criado:

- Incompatibilidade da ferramenta de teste com a linguagem ou IDE utilizada;
- Falta de conexão com a Internet via rede Wi-Fi.

## 8. METODOLOGIA DE TESTES

A metodologia adotada neste plano de testes foi o DDD (Domain-Driven Design), promovendo uma abordagem estruturada para garantir a qualidade e alinhamento do sistema com os requisitos.

- DDD (Domain-Driven Design): O sistema será desenvolvido com base em modelos de domínio bem definidos, centrando-se na lógica do negócio. Essa abordagem facilita a criação de testes que validam diretamente os comportamentos do domínio, promovendo testes alinhados ao funcionamento real da aplicação.

## 9. ABORDAGEM DOS TESTES

A abordagem de testes será dividida em três níveis principais, sendo:

- Testes de Unidade: Focados na validação de métodos e funções isoladas, com uso do framework xUnit. Estes testes garantem que os componentes internos do sistema estejam funcionando conforme o esperado.
- Testes Funcionais: Serão realizados utilizando o Selenium e DevTools, com o objetivo de validar se as funcionalidades implementadas atendem aos requisitos descritos.
- Testes Não Funcionais: Serão conduzidas avaliações com o uso de ferramentas como Lighthouse e SonarQube.

## 10. RESULTADOS DOS TESTES