

# Plano de Testes

## Sistema de Gerenciamento de Cafeteria

Alessandro Bento;  
Jefferson Cunha;  
João Cunha;  
Rafael Schmidt;

27/02/2024 - 27/02/2024

<b>Sistema de Gerenciamento de Cafeteria</b>	<b>2</b>
<b>1. INTRODUÇÃO</b>	<b>4</b>
<b>2. OBJETIVOS E TAREFAS</b>	<b>5</b>
2.1. Objetivos	5
2.2. Tarefas	5
<b>3. ESCOPO</b>	<b>6</b>
<b>4. Estratégia de teste</b>	<b>6</b>
4.1. Teste teste unitário	6
4.2. Teste de sistema e integração	7
4.3. Teste de desempenho e estresse	7
4.4. Testes de aceitação do usuário	7
4.5. Teste em lote/regressão Automatizados	7
<b>5. Requisitos de hardware</b>	<b>7</b>
<b>6. Requisitos de Software</b>	<b>7</b>
6.1. Ferramentas de Testes	7
<b>7. Calendários de testes</b>	<b>7</b>
<b>8. Procedimentos de Controle]</b>	<b>7</b>
<b>9. Recursos/Funções a serem testados</b>	<b>7</b>
<b>10. Recursos que não devem ser testados</b>	<b>7</b>
<b>11. Recursos/Papéis e Responsabilidades</b>	<b>7</b>
<b>12. Documentos/Evidências dos Testes</b>	<b>7</b>
<b>13. Dependências/Restrições</b>	<b>7</b>
<b>14. Riscos/Suposições</b>	<b>7</b>
<b>15. Aprovações</b>	<b>7</b>

# 1. INTRODUÇÃO

## Descrição Geral:

O Sistema de Gerenciamento de Cafeteria é uma aplicação projetada para facilitar a administração e operação da cafeteria. Ele oferece funcionalidades essenciais para otimizar o fluxo de trabalho e melhorar a experiência tanto dos clientes quanto dos funcionários.

## Principais Funções:

### 1. CRUD Produto:

Cadastro de Produtos com as seguintes informações: ID, Nome, Preço, Descrição, Imagem URL.

### 2. CRUD Comanda:

Cadastro de Comanda com as seguintes informações: ID, ID Pedido, ID Cliente, Número, Preço Total, Status.

### 3. CRUD Pedido:

Cadastro de Pedido com as seguintes informações: ID, ID Produto, Quantidade, Preço Pedido, Status.

### 4. CRUD de Clientes:

Cadastro simples e rápido dos clientes da cafeteria. Registra as informações de nome e número de telefone, e armazena o histórico de compras.

- Facilita o atendimento personalizado e o acompanhamento das preferências dos clientes.

### 5. Vendas e Pagamentos:

Realiza vendas de produtos aos clientes, aceita diferentes formas de pagamento, como dinheiro, cartões de crédito e cartões de consumo.

### 6. Cartões de Consumo(comanda):

Gerencia os cartões utilizados pelos clientes para comprar produtos, comandas.

### 7. Relatórios e Análises:

Gera relatórios sobre vendas, estoque, Ajuda na tomada de decisões estratégicas e no planejamento.

## 2. OBJETIVOS E TAREFAS

### 2.1. Objetivos

2.2.

### 2.3. Tarefas

#### **Cadastro de Funcionário**

CRUD de funcionários para controle de horas trabalhadas; **clientes que atendeu;** **quadro de horários.** Certifique-se de entender completamente o comportamento esperado de cada função.

#### **Cadastro de cliente**

Cadastrar os clientes para termos informações de consumo e frequência na cafeteria. CRUD de Cliente. Certifique-se de entender completamente o comportamento esperado de cada função.

#### **CRUD de produtos**

Incluir produtos novos no menu e controlar a disponibilidade de produtos. Certifique-se de entender completamente a função do registro dos produtos no sistema.

#### **Comanda e Pedido**

Ligar o cartão de comanda, ao com o cliente. Controle das comandas e dos pedidos nas mesmas.

Certifique-se de entender completamente o funcionamento esperado das comandas e pedidos .

## 3. ESCOPO

### 1. Dados de Teste:

- Prepare dados de teste realistas para simular o uso do sistema.
- Como fazer o cadastro do primeiro usuário.  
Dados das personagens fictícias para os testes.  
Planejar erros propositais e erros não propositais.
- Use dados fictícios para clientes, funcionários e produtos.

### 2. Execução dos Testes:

- Execute os cenários de teste conforme planejado.  
Anotar dados do momento que rodar o testes
- Registre os resultados, incluindo quaisquer problemas encontrados.

### 3. Relatórios e Documentação:

- Documente todos os testes realizados e seus resultados.
- Relate quaisquer problemas encontrados e sugerir correções.

## 4. Estratégia de teste

Para cada função, crie cenários de teste que abordam diferentes casos de uso.

### **Cliente**

Cadastro de Clientes:

Cenário 1: Registrar um novo cliente com informações válidas.

Cenário 2: Tentar registrar um cliente sem fornecer informações obrigatórias.

Cenário 3: Atualizar os detalhes de um cliente existente.

Repita esse processo para todas as funções.

### **Funcionário**

Cadastro Funcionario:

Cenário 1: Registrar um novo Funcionário com informações válidas.

Cenário 2: Tentar registrar um Funcionário sem fornecer informações obrigatórias.

Cenário 3: Atualizar os detalhes de um Funcionário existente.

Repita esse processo para todas as funções.

### **Pedido**

Cadastro Pedido:

Cenário 1: Registrar um novo Pedido com informações válidas.

Cenário 2: Tentar registrar um Pedido sem fornecer informações obrigatórias.

Cenário 3: Atualizar os detalhes de um Pedido existente.

Repita esse processo para todas as funções.

### **Comanda**

Cadastro Comanda:

Cenário 1: Registrar um novo Comanda com informações válidas.

Cenário 2: Tentar registrar um Comanda sem fornecer informações obrigatórias.

Cenário 3: Atualizar os detalhes de um Comanda existente.

Repita esse processo para todas as funções.

## Produto

Cadastro Produto:

Cenário 1: Registrar um novo Produto com informações válidas.

Cenário 2: Tentar registrar um Produto sem fornecer informações obrigatórias.

Cenário 3: Atualizar os detalhes de um Produto existente.

Repita esse processo para todas as funções.

### 4.1. Teste teste unitário

**Participantes:**

Testes Unitários - Responsáveis	
Nome	Equipe
Alessandro Bento	Desenvolvimento

**Metodologia:**

- Testes unitários escritos em Junit, após as classes em teste serem codificadas e testadas no desenvolvimento.
- Testes unitários escritos executados por classe e seguindo a estratégia de testes definida no item 4.0
- Registro dos erros/falhas nas classes e correção imediata no momento da ocorrência e/ou assim que possível, definido pendência até correção da falha encontrada.
- Deverão ser realizados tanto no Back-End.

### 4.2. Teste de sistema e integração

- Testes de Integração objetiva testar todas as funcionalidades Manter Clientes e Manter Comandas, Manter Pedidos, Manter Produtos, Manter Funcionario e seus relacionamentos.
- Deverão ser executados de forma automatizada e com repetições mínimas de 20 vezes por rodada de testes.
- Deverão verificar os relacionamentos das entidades e suas correspondências (integridade referencial de entidades Produto → Pedido → Comanda)
- Deverão ser realizados tanto no Back-End quanto no Front-End.

**Participantes:**

Testes Unitários - Responsáveis	
Nome	Equipe
Alessandro Bento	Desenvolvimento

#### Metodologia:

- Testes de integração executados na medida que as funcionalidades forem sendo desenvolvidas e liberadas e objetivam testar a integração de todos os componentes do sistema estão funcionais e de acordo com as especificações.
- Testes realizados de forma automatizada com Postman/Selenium Deverão seguir a estratégia geral definida no item 4.0
- Deverão executar rodadas de 10 repetições ou mais, abrangendo todas as estratégias e cobrindo 100% das funções do sistema.
- Deverão ser realizados tanto no Back-End quanto no Front-End.

### 4.3. Teste de desempenho e estresse

#### Participantes:

Testes Unitários - Responsáveis	
Nome	Equipe
Alessandro Bento	Desenvolvimento

#### Metodologia:

- Testes funcionais objetivam testar as funcionalidades do sistema, conforme definidas nos documentos de requisitos.
- Deverão testar todas as funções do sistema, devendo ser executados de forma automatizada e segmentados por funções/casos de uso do sistema.
- Registrar os erros/falhas no momento da ocorrência e/ou assim que possível, definido pendência até correção do problema.
- Deverão ser realizados tanto no Back-End quanto no Front-End.

Não foi possível realizar estes testes

### 4.4. Testes de aceitação do usuário

### Participantes:

Testes Unitários - Responsáveis	
Nome	Equipe
Alessandro Bento	Desenvolvimento

### Metodologia:

- Os usuários indicados pelos gestores do sistema CoffeeBrew, deverão ser cadastrados no sistema, e uma massa de dados deve ser inserida, objetivando as verificações iniciais das funcionalidades.
- A execução dos testes de Aceitação do Usuário, serão realizadas pelos usuários indicados. Estes deverão proceder o acesso a todas as funcionalidades do sistema, desenvolvidas em conformidade com os requisitos funcionais (casos de uso, diagrama de casos de uso), e não funcionais, como tempo de resposta das funções e desempenho geral do sistema.
- Deverão atestar a aceitação ou não das funcionalidades do sistema, ao todo ou em partes, indicando assim a aceitação total ou parcial do sistema.
- Nos casos de não aceitação parcial ou total do sistema, os gestores/usuários deverão indicar as inconformidades à equipe de desenvolvimento/testes para possíveis correções e/ou ajustes.
- Essa verificação de conformidade para aceitação ou rejeição do sistema, deve ser feita à luz dos requisitos funcionais e não funcionais, os quais foram definidos pelos gestores e documentados, analisados e modelados pela equipe de TI/Desenvolvimento.

## 5. Requisitos de hardware

Computadores para os testadores, servidores de testes para o sistema e banco de dados, conexão com a Internet e rede de computadores interligadas.

## 6. Requisitos de Software

Deverão ser utilizadas as seguintes ferramentas para a execução das atividades de testes definidas neste Plano de Testes:

Testes Unitários: Junit, versão 5.10.2

Testes de Regressão/Unitários: Junit, versão 5.10.2,

Testes de Integração/Regressão: Selenium IDE, versão 3.17.2, Postman, versão 11.2.0

Testes de Aceitação do Usuário, utilizar o próprio sistema



## 7. Procedimentos de Controle

### Relatório de problemas

Documentar os procedimentos a serem seguidos quando um incidente for encontrado durante o processo de teste. Utilizar um documento texto ou uma planilha e anexe uma cópia **“Apêndice” ao Plano de Testes**. Poderá ser utilizado um sistema para registro de incidentes/problemas encontrados ao executar os testes, ou então poder ser feito em uma planilha.

### Pedidos de mudança

As mudanças foram feitas diretamente no código

## 8. Recursos/Funções a serem testados

Recurso e/ou Funções	Testes
Clientes	CRUD- Cadastro, Consulta, Atualização, Exclusão, Listar
Produtos	CRUD- Cadastro, Consulta, Atualização, Exclusão, Listar
Comanda	Cadastro, Atualização, Consulta, Listar
Funcionário	CRUD- Cadastro, Consulta, Atualização, Exclusão, Listar
Pedido	CRUD- Cadastro, Consulta, Atualização, Exclusão, Listar

## 9. Recursos que não devem ser testados

Todos os recursos e aspectos do site será profundamente testado antes de ser lançado para o público

## 10. Recursos/Papéis e Responsabilidades

Recursos	Papel/Função	Responsável
Casos de testes de Aceitação	Usuária/Gestor	Alessandro

Realizar/executar os testes	Desenvolvedor/Tester	Alessandro, Jefferson, João, Rafael
Projetar/definir rotinas de testes	Analista/Desenvolvedor	Alessandro
Realizar/executar testes de aceitação	Usuário/Gestor	Alessandro
Ambiente de testes/Servidores, etc	Analistas/Suporte	Alessandro
Problemas/Contingências	Gerente TI	Alessandro

## 11. Documentos/Evidências dos Testes

Documentos de entrega deste Plano de Testes

- Plano de teste -> Este documento.
- Casos de teste
- Relatórios de incidentes (erros/falhas) de teste (relatório ou planilha)
- Relatórios de resumo de teste e/ou evidências dos testes (telas, logs, etc).

## 12. Dependências/Restrições

Caso aconteçam restrições significativas para a execução dos testes como indisponibilidade de itens de teste e outros recursos computacionais ou Software de teste, estas restrições/problemas devem ser reportados neste item do Plano de Testes.

## 13. Riscos/Suposições

Caso ocorram atrasos nas entregas dos itens (programas) a serem testados e/ou falta de ferramenta de testes, devem ser relatados aqui, indicando possíveis graus de risco para a entrega dos testes, conforme planejamento, bem como as contingências para cada caso.

<b>Recurso</b>	<b>Grau de Risco</b>	<b>Contingência</b>
Falta de Itens de teste	baixo	Reagendar testes e/ou antecipar testes em outros itens
Falta de software de teste	médio	Buscar alternativas de softwares para a mesma função
Falta de recursos computacionais	alta	Procurar junto aos gestores, alternativas e ou possíveis aquisições.

## 14. Aprovações

<b>APROVAÇÕES</b>		
<b>Nome/Responsável</b>	<b>Assinatura</b>	<b>Data</b>
Alessandro Bento		__/__/__
Jefferson Cunha		__/__/__
João Cunha		__/__/__
Rafael Schimidt		__/__/__