

Plano de Testes

[IFood]

Versão	Data	Alterações	Fonte	Autoria
1.0	21/05/2025	Definição testes unitários (Lucas)	Lucas	Lucas
1.0	21/05/2025	Definição testes unitários (Lucas)	Lucas	Lucas
1.0	21/05/2025	Definição testes unitários (Fernando)	Fernando	Fernando
1.0	21/05/2025	Definição testes unitários (Fernando)	Fernando	Fernando
1.0	22/05/2025	Definição de testes de integração, sistemas e aceitação (Lucas)	Lucas	Lucas
1.0	22/05/2025	Definição de testes de integração (Fernando)	Fernando	Fernando
1.0	22/05/2025	Definição de testes de sistemas e aceitação (Fernando)	Fernando	Fernando
2.0	24/05/2025	Desenvolvimento da documentação	Lucas/Fernando	Lucas/Fernando
2.0	28/05/2025	Desenvolvimento da documentação	Lucas/Fernando	Lucas/Fernando

SUMÁRIO

1.	INTRODUÇÃO	4
2.	SISTEMA QUE SERÁ TESTADO	4
3.	REQUISITOS A SEREM TESTADOS	4
4.	PROJETO DE TESTES	4
5.	CENÁRIOS DE TESTE	5
5.1.	Título do cenário de teste // <i>uma seção para cada cenário de teste</i>	5
6.	CRONOGRAMA	6
7.	CONCLUSÃO	7

1. INTRODUÇÃO

A ClickTest é formada por Lucas e Fernando, com foco em qualidade e em entregar soluções mais confiáveis para os usuários. Nosso objetivo como equipe é aplicar boas práticas de testes para garantir que os sistemas atendam às expectativas de quem usa e de quem desenvolve.

Este documento apresenta o plano de testes elaborado para uma aplicação web voltada ao serviço de entregas. Aqui, organizamos os diferentes tipos de testes, buscando validar desde funcionalidades específicas até a experiência geral de uso. A intenção é garantir que o sistema esteja funcionando corretamente, seja fácil de usar e atenda aos requisitos propostos.

2. SISTEMA QUE SERÁ TESTADO

O sistema iFood Web é uma plataforma online voltada à realização de pedidos de comida por meio de navegador. Seu principal objetivo é conectar usuários a uma grande variedade de restaurantes e estabelecimentos, permitindo uma experiência prática e rápida de compra de refeições, bebidas e outros produtos alimentícios.

A versão web do iFood oferece uma interface intuitiva, onde é possível realizar diferentes ações relacionadas ao processo de pedido e acompanhamento. Para este plano de testes, o foco será nos seguintes módulos principais da plataforma:

1 - Autenticação de usuário: Responsável por permitir que usuários se cadastrem e acessem suas contas utilizando e-mail e senha, garantindo a segurança e personalização da experiência.

2 - Busca e navegação de restaurantes: Nessa funcionalidade, os usuários podem pesquisar por estabelecimentos com base em filtros como tipo de comida, distância, promoções e avaliação. Também é possível visualizar o cardápio de cada restaurante.

3 - Carrinho de compras e personalização de pedidos: Aqui, os usuários podem adicionar itens ao carrinho, selecionar opções adicionais (como acompanhamentos ou retirada de ingredientes) e revisar o pedido antes de finalizar.

4 - Finalização do pedido e pagamento: O sistema permite ao usuário escolher a forma de pagamento (como cartão de crédito, débito ou Pix), definir endereço de entrega e confirmar a compra.

5 - Acompanhamento de pedidos em tempo real: Após a confirmação do pedido, o usuário pode acompanhar o status da entrega, desde a preparação até a chegada do entregador.

3. REQUISITOS E REGRAS DE NEGÓCIO A SEREM TESTADOS

[Essa seção apresenta a lista dos requisitos que serão testados nesse projeto de testes. Para cada requisito, destacar sua prioridade (a prioridade pode ser calculada a partir do cálculo do impacto (1,3,5,9) x probabilidade (1,3,5,9))]

Regras de negócio:

coloca as regras em tópicos aqui

- **RN01: Política de Complexidade de Senha**
- **RN02: Indexação de Busca**
- **RN03: Verificação de Estoque no Carrinho ao Finalizar Compra**
- **RN04: Validação e Cobertura de CEP no iFood**
- **RN05: Verificação de Disponibilidade de Item/Restaurante na Sacola do iFood**

ID	Descrição do requisito	Impacto	Probabilidade	Prioridade
RF01	Manter usuário	9	3	27
RF02	Gerenciamento de Senha do Usuário	9	3	27
RF03	Funcionalidade de Busca de Produtos	9	9	81
RF04	Carrinho de Compras Persistente	5	5	25
RF05	Aplicar Cupom de Desconto no Carrinho	9	5	45
RF06	Paginação de Resultados de Busca	3	5	15
RF07	Gerenciamento de Endereços de Entrega no iFood	9	9	81
RF08	Sistema de Notificações do iFood	5	9	45

RF09	Agendar um Pedido no iFood	5	3	15
RF10	Verificação de Estoque no Carrinho ao Finalizar Compra	9	5	45
RF11	Gestão do Carrinho de Compras	9	5	45
RF12	Restringir múltiplos estabelecimentos por carrinho	5	3	15
RF13	Gerar código para pagamento via Pix	9	5	45
RF14	O sistema deve permitir que o usuário edite seu endereço cadastrado, atualizando corretamente os dados no banco de dados.	5	5	25
RF15	O sistema deve permitir que o usuário adicione e remova produtos do carrinho de compras, mantendo o estado atualizado no banco de dados.	9	5	45
RF16	O sistema deve permitir a criação e exclusão de contas de usuário, garantindo que essas operações sejam corretamente refletidas no banco de dados.	3	9	27
RF17	Efetuar Pedido	5	3	15

4. PROJETO DE TESTES

[Listagem dos níveis de testes sob os quais o sistema será testado, e para cada nível: as técnicas de testes, os critérios de seleção de entrada e as ferramentas que serão aplicadas]

Nível de Teste	Unitário
Técnicas de Teste	Caixa preta
Critérios de Seleção de Entradas	Análise do valor limite Baseado em casos de uso Tabela de Decisão Partição de Equivalência (Valores corretos)
Ferramentas	JUnit

Nível de Teste	Integração
Técnicas de Teste	Caixa preta

Critérios de Seleção de Entradas	Tabela de Decisão
	Baseado em Casos de Uso
	Partição de Equivalência (Valores corretos)
Ferramentas	Postman

Nível de Teste	Sistema
Técnicas de Teste	Caixa preta
Critérios de Seleção de Entradas	Tabela de Decisão
	Análise de valor limite
	Baseado em Casos de Uso
	Particionamento de Equivalência
Ferramentas	Selenium

Nível de Teste	Aceitação
Técnicas de Teste	Caixa preta
Critérios de Seleção de Entradas	Análise de Valor Limite
	Baseado em Casos de Uso
Ferramentas	SUS – System Usability Scale

Papel no Projeto de Testes	Responsabilidades	Colaborador
Gerente de Teste	<ul style="list-style-type: none">● <i>Elaborar Plano de Testes</i>● <i>Fornecer direção técnica</i>● <i>Gerir recursos</i>	Lucas/ Fernando
Arquiteto de Testes	<ul style="list-style-type: none">● <i>Escolher as ferramentas a serem utilizadas</i>● <i>Configurar o ambiente de testes</i>● <i>Treinar a equipe de testes</i>	Lucas/ Fernando
Analista de Testes	<ul style="list-style-type: none">● <i>Elaborar casos de testes</i>● <i>Executar testes de sistema</i>	Lucas/ Fernando
Testador	<ul style="list-style-type: none">● <i>Executar testes</i>● <i>Registrar resultados</i>	Lucas/ Fernando

	<ul style="list-style-type: none"> • <i>Recuperar-se de erros</i> • <i>Documentar defeitos</i> 	
Analista de Sistemas	<ul style="list-style-type: none"> • <i>Elaborar requisitos do software</i> 	Lucas/ Fernando
Programador	<ul style="list-style-type: none"> • <i>Implementar requisitos do software</i> • <i>Elaborar testes unitários</i> • <i>Elaborar testes de integração</i> 	Lucas/ Fernando

5. CENÁRIOS DE TESTE

Lucas F. Jacinto:

<i>RF01</i>	<i>Manter Usuário</i>
<i>RT11</i>	<i>O sistema deve permitir criar usuário informando todos os dados obrigatórios</i>
Objetivo do roteiro de teste	<i>Encontrar falhas na criação de um novo usuário do sistema.</i>
Localização	<i>UserController;</i>
Pré-condições	<i>Usuário não estar gravado no sistema</i>
Passo a passo	<i>Informar Email ou Celular</i> <i>Informar código recebido</i> <i>Informar nome completo</i> <i>Informar CPF</i>
Nível de teste	<i>Teste Unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Baseado em Casos de Uso</i> <i>Partição de Equivalência</i>
<i>CT01</i>	<i>Criar um usuário informando seus dados</i>
Entradas	<i>Email: "<u>lfeliciojacinto@gmail.com</u>"</i>

	<i>Telefone:"47 99110 - 6059"</i> <i>Código:"xxxxxx"</i> <i>Nome Completo:"Lucas Felicio Jacinto"</i> <i>CPF*:"10879958928"</i>
Resultado Esperado	<i>Instância de usuário é criada</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Ao final da execução do teste, o sistema deverá ter registrado um novo usuário com todos os dados fornecidos</i>
CT02	<i>Criar um usuário informando o código errado.</i>
Entradas	<i>Email: "<u>Ifeliciojacinto@gmail.com</u>"</i> <i>Telefone:"47 99110 - 6059"</i> <i>Código:"xxxxxx"</i>
Resultado Esperado	<i>Mensagem: "Código informado inválido"</i>
Prioridade	<i>Alta</i>
Pré-condições	<i>Sistema exibe mensagem de erro</i>
CT03	<i>Cadastrar um usuário ja existente</i>
Entrada	<i>Email: "<u>Ifeliciojacinto@gmail.com</u>"</i> <i>Telefone:"47 99110 - 6059"</i> <i>Código:"xxxxxx"</i> <i>Nome Completo:"Lucas Felicio Jacinto"</i> <i>CPF*:"10879958928"</i>
Resultado Esperado	<i>Mensagem:"Este email ja está vinculado a uma conta"</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Sistema impede a criação de usuário exibindo mensagem de erro</i>

RT12	<i>O sistema deve permitir excluir o perfil de um usuário</i>
Objetivo do roteiro de teste	<i>Realizar a exclusão do perfil de usuário do sistema</i>
Localização	<i>UserController</i>
Pré-condições	<i>Usuário deve estar cadastrado</i>
Passo a passo	<i>Informar motivo informar detalhes Excluir perfil</i>
Nível de teste	<i>Teste unitário</i>
Técnica de teste	<i>Caixa preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso</i>
CT04	<i>Excluir usuário</i>
Entrada	<i>Motivos: "Casos de teste" Detalhes: "Casos de teste"</i>
Resultado Esperado	<i>Mensagem: "Sua conta foi excluída com sucesso"</i>
prioridade	<i>Alta</i>
Pós-condições	<i>Sistema exibe mensagem de sucesso ao excluir mensagem</i>
RF11	<i>Gestão do Carrinho de Compras</i>
RT13	<i>O sistema deve controlar o carrinho de compras do usuário, limitando a quantidade de itens, verificando a disponibilidade dos produtos, mantendo os itens por um tempo determinado e calculando o valor total corretamente.</i>
Objetivo do roteiro de teste	<i>Verificar o comportamento do sistema ao gerenciar itens no carrinho</i>
Localização	<i>CarrinhoController.java</i>

Pré-condições	<i>Usuário deve estar logado.</i> <i>Deve existir pelo menos um restaurante aberto e um fechado no sistema.</i> <i>Itens cadastrados corretamente no cardápio dos restaurantes.</i>
Passo a passo	<i>Adicionar itens ao carrinho até o limite permitido.</i> <i>Tentar adicionar mais itens além do limite.</i> <i>Aguardar o tempo máximo configurado para manter os itens no carrinho.</i> <i>Adicionar item de restaurante fechado ao carrinho.</i> <i>Adicionar item de restaurante aberto ao carrinho.</i> <i>Verificar se o valor total do carrinho está correto com base nos itens e complementos.</i>
Nível de teste	<i>Teste Unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Baseado em Casos de Uso</i> <i>Análise do valor limite</i>
CT05	<i>Itens máximos que carrinho aceita</i>
Entrada	<i>Produto: "Pastel"</i> <i>Quantidade: "1000"</i>
Resultado Esperado	<i>Mensagem: "Ops! Você atingiu o limite máximo de itens no carrinho"</i>
prioridade	<i>Alta</i>

Pós-condições	<i>Sistema exibirá mensagem de impedimento ao usuário realizar a ação</i>
CT06	<i>Tempo que item fica no carrinho</i>
Entrada	<i>Produto:"Bolo"</i> <i>Quantidade:"1"</i>
Resultado Esperado	<i>O sistema mantém o item no carrinho até que ocorra um impedimento, como:</i> <ul style="list-style-type: none"> • <i>o restaurante encerrar o atendimento,</i> • <i>o item se tornar indisponível,</i> • <i>o carrinho ultrapassar o tempo-limite configurado (ex: 45 minutos),</i> • <i>ou o usuário remover o item manualmente.</i>
prioridade	<i>Média</i>
Pós-condições	<i>O sistema faz a remoção do item do carrinho.</i>
CT07	<i>Adicionar itens de estabelecimentos fora do horário de atendimento</i>
Entrada	<i>Produto:"Açaí"</i> <i>Quantidade:"1"</i>
Resultado Esperado	<i>Mensagem:"Esta loja está fechado no momento."</i>
prioridade	<i>Alta</i>
Pós-condições	<i>Sistema impede adição de item ao carrinho</i>
CT08	<i>Somar Valor dos itens no carrinho</i>
Entrada	<i>Produto:"Hambúrguer"</i> <i>Quantidade:"1"</i> <i>Produto:"Pizza"</i>

	<i>Quantidade:"2"</i>
Resultado Esperado	<i>O sistema deve exibir corretamente o valor total do carrinho</i>
prioridade	<i>Alta</i>
Pós-condições	<i>O carrinho permanece atualizado com o valor total correto e refletindo qualquer alteração</i>
RF12	<i>Restringir múltiplos estabelecimentos por carrinho</i>
RT14	<i>Verificar restrição de itens de diferentes estabelecimentos no carrinho</i>
Objetivo do roteiro de teste	<i>Garantir que o sistema bloqueie a adição de itens de restaurantes diferentes no mesmo carrinho e informe o usuário adequadamente.</i>
Localização	<i>Carrinhocontroller.adicionarItem()</i>
Pré-condições	<i>O usuário deve estar logado.</i> <i>O carrinho deve conter pelo menos um item de um restaurante A.</i> <i>Deve existir um restaurante B (diferente do A) com itens disponíveis.</i>
Passo a passo	<i>Adicionar um item do restaurante A ao carrinho.</i> <i>Navegar até o cardápio do restaurante B.</i> <i>Tentar adicionar um item do restaurante B ao carrinho.</i> <i>Verificar se o sistema</i>
Nível de teste	<i>Teste unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso</i>
CT09	<i>Impedir usuário de adicionar itens de dois estabelecimentos diferentes ao carrinho</i>
Entrada	<i>"primeiroItem": {</i>

	<pre> "Produto": "Pizza Calabresa", "quantidade": 1 }, "segundoItem": { "Produto": "Hambúrguer", "quantidade": 1 } } </pre>
Resultado Esperado	<i>Mensagem: "Voce só pode adicionar itens de uma loja por vez"</i>
prioridade	<i>Alta</i>
Pós-condições	<i>Sistema exibe mensagem sobre o ocorrido e apresenta a opção de:</i> <ul style="list-style-type: none"> • <i>Esvaziar sacola e adicionar</i> • <i>Cancelar</i>
RF13	<i>Gerar código para pagamento via Pix</i>
RT15	<i>O sistema deve gerar automaticamente um código Pix válido quando o usuário selecionar essa forma de pagamento.</i>
Objetivo do roteiro de teste	<i>Verificar se o sistema gera corretamente o código QR ou copia/cola Pix ao selecionar essa forma de pagamento.</i>
Localização	<i>Pagamentocontroller.gerarCodigoPix()</i>
Pré-condições	<i>Usuário logado.</i> <i>Carrinho com itens válidos.</i> <i>Forma de pagamento Pix selecionada.</i> <i>API de geração de QR Code ou Pix ativa.</i>
Passo a passo	<i>Preencher o carrinho normalmente.</i> <i>Ir para a tela de pagamento.</i> <i>Selecionar a opção "Pix".</i>

	<p><i>Confirmar a seleção.</i></p> <p><i>Verificar se o sistema gera:</i></p> <ul style="list-style-type: none"> • <i>Código Pix (copiar e colar)</i> • <i>QR Code visível e válido</i>
Nível de teste	<i>Teste unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso</i>
CT10	<i>Pagamento via Pix</i>
Entrada	<i>FormaPagamento:"Pix"</i>
Resultado Esperado	<i>Sistema gera código copia e cola para pagamento do pedido</i>
prioridade	<i>Alta</i>
Pós-condições	<i>Após o sistema validar o pagamento, informar que o mesmo foi validado e apresenta o acompanhamento do pedido</i>
RF14	<i>O sistema deve permitir que o usuário edite seu endereço cadastrado, atualizando corretamente os dados no banco de dados.</i>
RT16	<i>Edição de Endereço de Usuário no Sistema</i>
Objetivo do roteiro de teste	<i>Verificar a integração entre o módulo de edição de dados do usuário e o serviço de atualização de endereço.</i>
Localização	<i>EnderecoController.updateEndereco()</i>
Pré-condições	<i>Usuário autenticado com um endereço previamente cadastrado.</i>
Passo a passo	<p><i>Informar novos dados do endereço</i></p> <p><i>Confirmar edição</i></p>
Nível de teste	<i>Teste de Integração</i>

Técnica de teste	<i>Caixa-preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso</i>
CT11	<i>Editar endereço</i>
Entrada	<i>numero:"544" Complemento:"Casa"</i>
Resultado Esperado	<i>Endereço alterado é cadastrado no banco de dados</i>
prioridade	<i>Alta</i>
Pós-condições	<i>O ficará com o endereço alterado salvo no banco de dados</i>
RF15	<i>O sistema deve permitir que o usuário adicione e remova produtos do carrinho de compras, mantendo o estado atualizado no banco de dados.</i>
RT17	<i>Adição e Remoção de Itens do Carrinho de Compras</i>
Objetivo do roteiro de teste	<i>Verificar se o serviço de carrinho se integra corretamente com o serviço de catálogo e sessão do usuário.</i>
Localização	<i>Carrinhocontroller</i>
Pré-condições	<i>Usuário autenticado e logado em uma sessão ativa</i>
Passo a passo	<i>Adicionar produtos ao carrinho de compras remover produtos do carrinho de compras</i>
Nível de teste	<i>Teste de Integração</i>
Técnica de teste	<i>Caixa-preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso</i>
CT12	<i>Adicionar/ remover itens do carrinho</i>
Entrada	<i>Produto:"Pizza" Quantidade:"1"</i>

Resultado Esperado	<i>O fluxo de adicionar e remover ocorre normalmente</i>
prioridade	<i>Alta</i>
Pós-condições	<i>O sistema não possuirá os produtos do carrinho cadastrados no banco</i>
RF16	<i>O sistema deve permitir a criação e exclusão de contas de usuário, garantindo que essas operações sejam corretamente refletidas no banco de dados.</i>
RT18	<i>Verificar criação e exclusão de usuário</i>
Objetivo do roteiro de teste	<i>Verificar se o sistema realiza corretamente a criação e a exclusão de contas de usuário, garantindo que os dados sejam devidamente inseridos ou removidos do banco de dados.</i>
Localização	<i>UserController</i>
Pré-condições	<i>Sistema em execução e banco de dados acessível.</i> <i>Para o teste de exclusão, deve existir pelo menos um usuário cadastrado.</i> <i>Para o teste de criação, o usuário não pode estar cadastrado no sistema</i>
Passo a passo	<i>Criar usuário informando os dados corretos</i> <i>Excluir usuário</i>
Nível de teste	<i>Teste de integração</i>
Técnica de teste	<i>Caixa Cinza</i>
Critério de seleção de entradas	<i>Baseado em casos de uso</i>
CT13	<i>Excluir usuário</i>
Entrada	<i>Motivos: "Casos de teste"</i> <i>Detalhes: "Casos de teste"</i>
Resultado Esperado	<i>O usuário em questão é removido do banco de dados</i>

prioridade	<i>Alta</i>
Pós-condições	<i>O sistema não apresenta mais o usuário cadastrado</i>
CT14	Criar conta colocando todos os dados
Entrada	<i>Email: "<u>lfeliciojacinto@gmail.com</u>"</i> <i>Telefone: "47 99110 - 6059"</i> <i>Código: "xxxxxx"</i> <i>Nome Completo: "Lucas Felicio Jacinto"</i> <i>CPF: "10879958928"</i>
Resultado Esperado	O usuário é cadastrado no banco de dados
prioridade	Alta
Pós-condições	O sistema terá o usuário em questão inserido no banco de dados
RF01	<i>Manter Usuário</i>
RT19	Criação e Edição de Conta de Usuário
Objetivo do roteiro de teste	Verificar se o sistema permite criar uma nova conta de usuário e editar corretamente os dados cadastrados
Localização	UsuarioController
Pré-condições	Criação: Usuário deve estar cadastrado
Passo a passo	Editar: Usuário deve estar cadastrado Criar: Email: "lfeliciojacinto@gmail.com" Telefone: "47 99110 - 6059" Código: "xxxxxx" Nome Completo: "Lucas Felicio Jacinto" CPF*: "10879958928" Editar: Editar campos desejados

Nível de teste	Sistema
Técnica de teste	Caixa preta
Critério de seleção de entradas	Baseado em casos de uso
CT15	Criar usuário
Entrada	<i>Email: "<u>lfeliciojacinto@gmail.com</u>"</i> <i>Telefone: "47 99110 - 6059"</i> <i>Código: "xxxxxx"</i> <i>Nome Completo: "Lucas Felicio Jacinto"</i> <i>CPF: "10879958928"</i>
Resultado Esperado	Usuário é cadastrado no sistema e é direcionado a tela principal
prioridade	Alta
Pós-condições	Usuário estará logado na aplicação e registrado no banco de dados
CT16	Editar usuário
Entrada	Nome: "Lucas Jacinto" CPF: "10879958928"
Resultado Esperado	Dados do usuário atualizados com sucesso no perfil e mensagem: "Atualizado com sucesso"
prioridade	Alta
Pós-condições	Os dados do usuário no banco de dados foram atualizados com as novas informações assim como o usuário permanece logado.
RF17	Efetuar pedido
RT20	Finalização de Pedido pelo Usuário
Objetivo do roteiro de teste	Garantir que o usuário consiga finalizar corretamente um pedido com os itens do carrinho, confirmando endereço de entrega e método de pagamento.
Localização	PedidoController

Pré-condições	O usuário deve estar autenticado. O carrinho deve conter ao menos um item. Os dados de entrega e pagamento devem estar válidos.
Passo a passo	Acessar o carrinho de compras. Selecionar endereço de entrega. Escolher o método de pagamento. Confirmar e finalizar o pedido. Verificar geração do pedido no sistema.
Nível de teste	Sistema
Técnica de teste	Caixa Preta
Critério de seleção de entradas	Baseado em casos de uso
CT17	Realizar pedido
Entrada	Produto:"Pizza" Quantidade:"1" Forma de pagamento:"Pix"
Resultado Esperado	Após pedido aprovado, apresentar tela de acompanhamento do mesmo
prioridade	Alta
Pós-condições	Apresentará a tela para acompanhamento do pedido
RF15	<i>O sistema deve permitir que o usuário adicione e remova produtos do carrinho de compras, mantendo o estado atualizado no banco de dados.</i>
RT21	Manipulação de Itens no Carrinho de Compras
Objetivo do roteiro de teste	Validar se o sistema permite adicionar e remover produtos do carrinho, mantendo o estado atualizado e refletido corretamente no banco de dados.
Localização	CarrinhoController
Pré-condições	O usuário deve estar autenticado. Produtos devem estar disponíveis no sistema.
Passo a passo	Adicionar item: Acessar lista de produtos. Selecionar um produto e clicar em "Adicionar ao carrinho". Verificar se o item aparece no carrinho.

Nível de teste	Remover item: 4. Acessar o carrinho. 5. Clicar em “Remover” no item desejado. 6. Confirmar se o item foi removido e o estado do carrinho atualizado. Sistema
Técnica de teste	Caixa preta
Critério de seleção de entradas	Baseado em casos de uso
CT18	Adicionar/ remover itens do carrinho
Entrada	<i>Produto: “Pizza”</i> <i>Quantidade: “1”</i>
Resultado Esperado	Ao adicionar um item ao carrinho, ele é exibido e o subtotal é atualizado assim como ao remover um item, ele é retirado do carrinho e o subtotal é recalculado.
prioridade	Alta
Pós-condições	O sistema com as operações realizadas e o subtotal recalculado
RF17	Efetuar Pedido
RT22	Conclusão de Pedido com Sucesso
Objetivo do roteiro de teste	Validar se o sistema permite a finalização de um pedido com todos os dados necessários (itens no carrinho, endereço e pagamento) corretamente informados, garantindo que o pedido seja registrado e processado com sucesso.
Localização	PedidoController
Pré-condições	Usuário autenticado Carrinho contendo pelo menos um item Endereço de entrega e método de pagamento cadastrados e válidos
Passo a passo	Acessar o sistema com usuário autenticado Adicionar itens ao carrinho Acessar o carrinho e clicar em “Finalizar Pedido” Escolher o endereço de entrega Selecionar o método de pagamento

Nível de teste	Confirmar o pedido Verificar se o sistema retorna a mensagem de sucesso e armazena o pedido no banco de dados Aceitação
Técnica de teste	Caixa-preta
Critério de seleção de entradas	Baseado em casos de uso
CT19	Finalizar Pedido com Sucesso
Entrada	Produto:"Pastel" Quantidade:"1" Bairro:"Bela Vista" Rua:"José Bonifácio" Número:"544" Forma de pagamento:"Pix"
Resultado Esperado	O sistema confirma o pagamento e encaminha o usuário para acompanhamento do produto
prioridade	Alta
Pós-condições	O pedido entra na fila de preparo e fica disponível para acompanhamento.
Resultado a ser obtido	Finalização do processo de pagamento e redirecionar o usuário para a tela de acompanhamento do pedido.
Classificação	Abaixo de 50: Usabilidade baixa Entre 50 e 70: Usabilidade aceitável, mas com espaço para melhorias Acima de 70: Usabilidade boa, indicando que o sistema é fácil de usar Acima de 85: Usabilidade excelente, indicando que a maioria dos usuários achará o sistema muito fácil de usar
Descrição dos resultados a serem obtidos	O usuário deve executar a finalização do pedido em no máximo 2 minutos
RF01	Manter usuário
RT23	Validação do Login com Credenciais Válidas

Objetivo do roteiro de teste	Verificar possíveis inconsistências na criação de um usuário
Localização	UsuarioController
Pré-condições	O usuário deve estar previamente cadastrado no sistema com login (e-mail) e senha válidos
Passo a passo	<p>O sistema deve estar online e acessível</p> <p>Abrir a aplicação e acessar a tela de login</p> <p>Informar o e-mail previamente cadastrado</p> <p>Informar a senha correspondente ao e-mail</p> <p>Clicar no botão “Entrar”</p> <p>Verificar se o sistema realiza o redirecionamento para a tela principal da área logada</p> <p>Confirmar a exibição do nome ou identificação do usuário logado</p>
Nível de teste	Aceitação
Técnica de teste	Caixa-preta
Critério de seleção de entradas	Baseado em casos de uso
CT20	Login com Credenciais Válidas
Entrada	email: " feliciojacinto@gmail.com " numero: "47 991106059" código: "xxxxxx"
Resultado Esperado	O sistema autentica o usuário e redireciona para a tela inicial do app.
prioridade	Alta
Pós-condições	Sessão de usuário iniciada e dados carregados na tela inicial.
Resultado a ser obtido	Tela inicial exibida com o apresentando as funcionalidades disponíveis

Classificação	Abaixo de 50: Usabilidade baixa Entre 50 e 70: Usabilidade aceitável, mas com espaço para melhorias Acima de 70: Usabilidade boa, indicando que o sistema é fácil de usar Acima de 85: Usabilidade excelente, indicando que a maioria dos usuários achará o sistema muito fácil de usar
Descrição dos resultados a serem obtidos	O sistema permitirá o acesso do usuário apresentando a interface principal e suas funcionalidades disponíveis

Fernando:

RF02: Gerenciamento de Senha do Usuário

RN01: Política de Complexidade de Senha

<i>RF02</i>	<i>Gerenciamento de Senha do Usuário</i>
<i>RT01</i>	<i>Validar a funcionalidade de alteração de senha do usuário, incluindo a aplicação da política de complexidade e a não reutilização de senhas antigas.</i>
Objetivo do roteiro de teste	<i>Validar a funcionalidade de alteração de senha do usuário, incluindo a aplicação da política de complexidade e a não reutilização de senhas antigas.</i>
Localização	<i>UserService.changePassword() / AuthModule.updateCredentials()</i>
Pré-condições	<i>Usuário autenticado no sistema.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Acessar a funcionalidade de alteração de senha.</i> <i>2. Informar senha atual válida.</i> <i>3. Informar nova senha "aB1@cDe".</i> <i>4. Confirmar nova senha "aB1@cDe".</i> <i>5. Submeter a alteração.</i>

Nível de teste	<i>Teste Unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Tabela de Decisão</i>

CT21	<i>Tentar alterar senha com menos de 8 caracteres.</i>
Entradas	<ul style="list-style-type: none"> • <i>Senha Atual: "SenhaAntiga123!"</i> • <i>Nova Senha: "aB1@cDe" (7 caracteres)</i> • <i>Confirmação Nova Senha: "aB1@cDe"</i>
Resultado Esperado	<i>sistema deve exibir uma mensagem de erro indicando que a senha não atende ao requisito mínimo de 8 caracteres. A senha do usuário não deve ser alterada.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>A senha do usuário permanece inalterada.</i>

CT22	<i>Tentar alterar senha sem letra maiúscula.</i>
Entradas	<ul style="list-style-type: none"> • <i>Senha Atual: "SenhaAntiga123!"</i> • <i>Nova Senha: "novasenha123@" (sem maiúscula)</i> • <i>Confirmação Nova Senha: "novasenha123@"</i>
Resultado Esperado	<i>O sistema deve exibir uma mensagem de erro indicando que a senha não atende à política de complexidade (falta letra maiúscula). A senha do usuário não deve ser alterada.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>A senha do usuário permanece inalterada.</i>

CT23	<i>Tentar alterar senha sem caractere especial.</i>
Entradas	<ul style="list-style-type: none"> • <i>Senha Atual: "SenhaAntiga123!"</i> • <i>Nova Senha: "NovaSenha123" (sem caractere especial)</i>

	<ul style="list-style-type: none"> Confirmação Nova Senha: "NovaSenha123"
Resultado Esperado	O sistema deve exibir uma mensagem de erro indicando que a senha não atende à política de complexidade (falta caractere especial). A senha do usuário não deve ser alterada.
Prioridade	Alta
Pós-condições	A senha do usuário permanece inalterada.

CT24	Alterar senha com sucesso atendendo a todos os critérios da política.
Entradas	<ul style="list-style-type: none"> Senha Atual: "SenhaAntiga123!" Nova Senha: "NovaSenha@2025" Confirmação Nova Senha: "NovaSenha@2025"
Resultado Esperado	O sistema deve confirmar a alteração da senha. O usuário deve ser capaz de logar com a nova senha.
Prioridade	Alta
Pós-condições	Senha do usuário alterada com sucesso.

RF03: Funcionalidade de Busca de Produtos

RN02: Indexação de Busca

RF03	Funcionalidade de Busca de Produtos
RT02	Validar a funcionalidade de busca de produtos, incluindo a relevância e a capacidade de encontrar produtos por diferentes termos.
Objetivo do roteiro de teste	Validar a funcionalidade de busca de produtos, incluindo a relevância e a capacidade de encontrar produtos por diferentes termos.
Localização	ProductService.searchProducts() / SearchModule.query()
Pré-condições	Existem produtos cadastrados no sistema com nomes, descrições e tags variadas.

Passo a passo	6. <i>Acessar a funcionalidade de alteração de senha.</i> 7. <i>Informar senha atual válida.</i> 8. <i>Informar nova senha "aB1@cDe".</i> 9. <i>Confirmar nova senha "aB1@cDe".</i> 10. <i>Submeter a alteração.</i>
Nível de teste	<i>Teste Unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Partição de Equivalência (Valores corretos)</i>
CT25	<i>Buscar produto por nome exato.</i>
Entradas	<ul style="list-style-type: none"> • <i>Termo de Busca: "Pão"</i>
Resultado Esperado	<i>O sistema deve retornar o "Pão" entre os primeiros resultados, idealmente o primeiro.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Lista de produtos retornada conforme esperado.</i>

CT26	<i>Buscar produto por palavra-chave presente na descrição.</i>
Entradas	<i>Termo de Busca: "crocante"</i>
Resultado Esperado	<i>O sistema deve retornar produtos cuja descrição contenha "crocante".</i>
Prioridade	<i>Média</i>
Pós-condições	<i>Lista de produtos retornada conforme esperado.</i>

CT27	<i>Buscar produto por tag associada.</i>
Entradas	<i>Termo de Busca: "promoção"</i>

Resultado Esperado	<i>O sistema deve retornar produtos que possuam a tag "promoção".</i>
Prioridade	<i>Média</i>
Pós-condições	<i>Lista de produtos retornada conforme esperado.</i>

RF04: Carrinho de Compras Persistente

<i>RF04</i>	<i>Carrinho de Compras Persistente</i>
<i>RT03</i>	<i>Validar a persistência do carrinho de compras após logout e login do usuário.</i>
Objetivo do roteiro de teste	<i>Validar a persistência do carrinho de compras após logout e login do usuário.</i>
Localização	<i>ShoppingCartService.loadCart() / SessionModule.handleCartPersistence()</i>
Pré-condições	<i>Usuário possui itens no carrinho.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Adicionar itens ao carrinho.</i> <i>2. Realizar logout.</i> <i>3. Realizar login com o mesmo usuário.</i> <i>4. Acessar o carrinho.</i>
Nível de teste	<i>Teste Unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso.</i>

<i>CT28</i>	<i>Verificar persistência do carrinho após logout e login.</i>
Entradas	<i>N/A (ações do usuário)</i>
Resultado Esperado	<i>Os itens previamente adicionados ao carrinho devem estar presentes após o login.</i>

Prioridade	<i>Alta</i>
Pós-condições	<i>Carrinho restaurado com sucesso.</i>

RN11: Verificação de Estoque no Carrinho ao Finalizar Compra

RF11	<i>Verificação de Estoque no Carrinho ao Finalizar Compra</i>
RT04	<i>Validar a verificação de estoque no momento da finalização da compra.</i>
Objetivo do roteiro de teste	<i>Validar a verificação de estoque no momento da finalização da compra.</i>
Localização	<i>OrderService.validateCartStock() / CheckoutModule.verifyStock()</i>
Pré-condições	<i>Usuário com itens no carrinho, um dos quais terá seu estoque zerado antes da finalização.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Usuário adiciona Produto A (10 unidades em estoque) e Produto B (5 unidades em estoque) ao carrinho.</i> <i>2. Outro processo/usuário compra todas as 5 unidades do Produto B.</i> <i>3. O usuário original tenta finalizar a compra com Produto A e Produto B no carrinho.</i>
Nível de teste	<i>Teste Unitário</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Análise do valor limite.</i>

CT29	<i>Tentar finalizar compra com item que ficou sem estoque.</i>
Entradas	<i>Carrinho contendo Produto A e Produto B; Estoque do Produto B = 0.</i>

Resultado Esperado	<i>O sistema deve notificar o usuário que o Produto B não está mais disponível em estoque. O Produto B deve ser removido do carrinho ou marcado como indisponível, impedindo a finalização da compra com ele.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Compra não finalizada, usuário notificado sobre item indisponível.</i>

CT30	<i>Finalizar compra com todos os itens em estoque.</i>
Entradas	<i>Carrinho contendo Produto A e Produto B; Estoque de ambos os produtos > 0.</i>
Resultado Esperado	<i>O sistema deve permitir que o usuário prossiga para as próximas etapas da finalização da compra.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Usuário direcionado para a etapa de pagamento/confirmação.</i>

RF02	<i>Gerenciamento de Senha do Usuário</i>
RT05	<i>Validar o fluxo completo de recuperação de senha do usuário no sistema web.</i>
Objetivo do roteiro de teste	<i>Validar o fluxo completo de recuperação de senha do usuário no sistema web.</i>
Localização	<i>Fluxo de "Esqueci minha senha" no sistema web.</i>
Pré-condições	<i>Usuário com itens no carrinho, um dos quais terá seu estoque zerado antes da finalização.</i>
Passo a passo	<ol style="list-style-type: none"> <i>Acessar a página de login do sistema web.</i> <i>Clicar no link/botão "Esqueci minha senha".</i> <i>Informar um e-mail cadastrado no sistema.</i>

7. Clicar em "Enviar link de recuperação".
8. Verificar a caixa de entrada do e-mail informado.
9. Clicar no link recebido.
10. Informar nova senha e confirmação, seguindo a política de senha do sistema.
11. Submeter a nova senha.
12. Tentar logar com a nova senha.

Nível de teste	<i>Sistema</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Baseado em casos de uso.</i>

CT31	<i>Solicitar recuperação de senha com e-mail válido.</i>
Entradas	<i>E-mail: (e-mail válido cadastrado), Nova Senha: (senha válida)</i>
Resultado Esperado	<i>Usuário recebe o e-mail com link para redefinição. Após redefinir, consegue logar com a nova senha. O link de recuperação deve ter um tempo de validade.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Senha do usuário alterada com sucesso. Usuário consegue acessar o sistema.</i>

RF05: Aplicar Cupom de Desconto no Carrinho

RF05	<i>Aplicar Cupom de Desconto no Carrinho</i>
RT06	<i>Validar a aplicação de cupons de desconto válidos e inválidos no carrinho de compras do sistema web.</i>
Objetivo do roteiro de teste	<i>Validar a aplicação de cupons de desconto válidos e inválidos no carrinho de compras do sistema web.</i>
Localização	<i>Página do Carrinho de Compras / Checkout no sistema web.</i>
Pré-condições	<i>Usuário logado com itens no carrinho. Existem cupons de desconto ativos e inativos/inválidos.</i>

Passo a passo

1. Acessar o carrinho de compras.
2. Localizar o campo para inserir código de cupom.
3. Inserir um código de cupom válido e ativo (e.g., "DESCONTO10").
4. Clicar em "Aplicar".

Nível de teste

Sistema

Técnica de teste

Caixa Preta

Critério de seleção de entradas

Tabela de decisão

CT32

Tentar aplicar cupom de desconto expirado/inválido.

Entradas

Código do Cupom: "DESCONTO10"

Resultado Esperado

O sistema deve aplicar o desconto correspondente ao cupom no subtotal do carrinho. O valor do desconto e o novo total devem ser exibidos corretamente.

Prioridade

Alta

Pós-condições

Desconto aplicado e refletido no total do carrinho.

CT33

Aplicar cupom de desconto válido.

Entradas

Código do Cupom: "CUPOMVELHO"

Resultado Esperado

O sistema deve exibir uma mensagem informando que o cupom é inválido, expirado ou não pôde ser aplicado. O total do carrinho não deve ser alterado.

Prioridade

Média

Pós-condições

Nenhuma alteração no carrinho. Mensagem de erro exibida.

RF07: Paginação de Resultados de Busca

RF07	Paginação de Resultados de Busca
RT07	<i>Validar a funcionalidade de paginação na tela de resultados de busca de produtos.</i>
Objetivo do roteiro de teste	<i>Validar a funcionalidade de paginação na tela de resultados de busca de produtos.</i>
Localização	<i>Página de resultados de busca de produtos no sistema web.</i>
Pré-condições	<i>Uma busca foi realizada e retornou um número de produtos maior que o limite de exibição por página.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Realizar uma busca que retorne múltiplos resultados (e.g., mais de 20 itens, se a página exibe 10 por vez).</i> <i>2. Verificar os itens da primeira página.</i> <i>3. Clicar no controle de paginação para "Exibir mais"</i>
Nível de teste	<i>Sistema</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Análise do valor limite.</i>

CT34	Navegar para a próxima página de resultados e verificar os itens.
Entradas	<i>N/A (interação com controle de paginação)</i>
Resultado Esperado	<i>O sistema deve exibir a segunda página de resultados, com itens diferentes da primeira página. Os controles de paginação (anterior, números de página) devem ser atualizados corretamente.</i>
Prioridade	<i>Média</i>
Pós-condições	<i>Segunda página de resultados exibida corretamente.</i>

RF08: Gerenciamento de Endereços de Entrega no iFood

RN04: Validação e Cobertura de CEP no iFood

RF08	Gerenciamento de Endereços de Entrega no iFood
RT08	<i>Validar a integração do cadastro de endereço no iFood com o serviço de validação de CEP e verificação de área de cobertura.</i>
Objetivo do roteiro de teste	<i>Validar a integração do cadastro de endereço no iFood com o serviço de validação de CEP e verificação de área de cobertura.</i>
Localização	<i>iFood.AddressManager.saveNewAddress() integra com iFood.GeocodingService.lookupCep() e iFood.CoverageService.checkArea()</i>
Pré-condições	<i>API de geolocalização/CEP e serviço de cobertura ativos.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Ir para "Meus Endereços" -> "Adicionar Novo".</i> <i>2. Digitar um CEP válido (ex: "01311-000" - Av. Paulista).</i> <i>3. (Sistema consulta serviço de CEP e cobertura).</i> <i>4. Preencher número e complemento.</i> <i>5. Salvar.</i>
Nível de teste	<i>Integração</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Partição de Equivalência (Valores corretos)</i>

CT35	Adicionar novo endereço no iFood com CEP válido e verificar autopreenchimento e cobertura.
Entradas	<i>CEP: "01311-000", Número: "1000", Complemento: "Bloco A"</i>
Resultado Esperado	<i>Campos de rua, bairro, cidade são autopreenchidos. Mensagem de "Área Atendida" (ou similar) é exibida. Endereço salvo.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Endereço adicionado com dados validados e dentro da área de cobertura.</i>

CT36	<i>Tentar adicionar novo endereço no iFood com CEP de área não coberta.</i>
Entradas	<i>CEP: "12345-000"</i>
Resultado Esperado	<i>Mensagem: "Infelizmente, ainda não atendemos esta região." ou similar. Não permite salvar o endereço para pedidos (pode permitir salvar para referência, mas não para uso).</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Usuário informado sobre a falta de cobertura.</i>

RF09: Sistema de Notificações do iFood

RN05: Verificação de Disponibilidade de Item/Restaurante na Sacola do iFood

RF09	<i>Sistema de Notificações do iFood</i>
RT09	<i>Validar a integração entre a mudança de status de um restaurante (ex: fecha inesperadamente) e o sistema de notificação ao usuário que possui itens desse restaurante na sacola.</i>
Objetivo do roteiro de teste	<i>Validar a integração entre a mudança de status de um restaurante (ex: fecha inesperadamente) e o sistema de notificação ao usuário que possui itens desse restaurante na sacola.</i>
Localização	<i>iFood.RestaurantStatusMonitor (altera status) integra com iFood.CartService.validateActiveRestaurants() que integra com iFood.PushNotificationService.sendAlert()</i>
Pré-condições	<i>Usuário logado com itens de um "Restaurante Y" na sacola.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Usuário adiciona "Temaki Salmão Completo" do "Restaurante Y" à sacola.</i> <i>2. (Simulação: "Restaurante Y" fica offline/fecha no sistema iFood).</i> <i>3. Sistema detecta a mudança de status do restaurante.</i>
Nível de teste	<i>Integração</i>

Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Partição de Equivalência (Valores corretos)</i>

CT37	<i>Notificar usuário via push quando restaurante com itens na sacola fecha inesperadamente.</i>
Entradas	<i>Itens do "Restaurante Y" na sacola; Status do "Restaurante Y" muda para "Fechado".</i>
Resultado Esperado	<i>O usuário recebe uma notificação push: "O Restaurante Y fechou. Infelizmente, seus itens foram removidos da sacola." Itens do "Restaurante Y" são removidos da sacola ou marcados como indisponíveis.</i>
Prioridade	<i>Média</i>
Pós-condições	<i>Usuário notificado. Sacola atualizada.</i>

CT38	<i>Alterar endereço de entrega durante o checkout e verificar recalcule da taxa de entrega e disponibilidade do restaurante.</i>
Entradas	<i>Mudança de endereço selecionado no checkout.</i>
Resultado Esperado	<i>A taxa de entrega é atualizada para o "Endereço Trabalho" (ex: R\$10,00 ou "Este restaurante não entrega neste endereço"). A lista de itens e o total são ajustados conforme.</i>
Prioridade	<i>Alta</i>
Pós-condições	<i>Taxa de entrega e disponibilidade para o novo endereço corretamente atualizadas.</i>

RF10: Agendar um Pedido no iFood

RF10	<i>Agendar um Pedido no iFood</i>
-------------	-----------------------------------

RT10	<i>Validar a funcionalidade de agendamento de pedido para um horário/data futura.</i>
Objetivo do roteiro de teste	<i>Validar a funcionalidade de agendamento de pedido para um horário/data futura.</i>
Localização	<i>Aplicativo iFood / Site iFood (em restaurantes que oferecem agendamento).</i>
Pré-condições	<i>Usuário logado. Restaurante selecionado oferece a opção de agendamento.</i>
Passo a passo	<ol style="list-style-type: none"> <i>1. Abrir o iFood.</i> <i>2. Encontrar um restaurante que permita agendamento (ex: "Padaria Pão Quente").</i> <i>3. Adicionar itens à sacola (ex: Pães, Frios).</i> <i>4. Na sacola, procurar e selecionar a opção "Agendar pedido".</i> <i>5. Selecionar data (dia seguinte) e horário disponível para entrega.</i> <i>6. Confirmar o agendamento.</i> <i>7. Selecionar forma de pagamento.</i> <i>8. Finalizar o agendamento.</i> <i>9. Verificar em "Meus Pedidos" se o pedido agendado aparece corretamente.</i>
Nível de teste	<i>Sistema</i>
Técnica de teste	<i>Caixa Preta</i>
Critério de seleção de entradas	<i>Análise do valor limite</i>

CT39	<i>Agendar um pedido para o dia seguinte em um restaurante que permite agendamento.</i>
Entradas	<i>Restaurante com agendamento, Itens, Data/Hora futura, Forma de Pagamento.</i>
Resultado Esperado	<i>O pedido é registrado como agendado para a data/hora selecionada. O usuário recebe uma confirmação do agendamento. O pedido não é enviado imediatamente ao restaurante, mas sim no momento programado.</i>
Prioridade	<i>Alta</i>

Pós-condições *Pedido agendado e visível no histórico com status "Agendado".*

CT40 *Tentar agendar pedido para um horário limite que se torna indisponível durante a seleção.*

Entradas *Restaurante com agendamento, Itens, Data/Hora futura (limite e que se torna indisponível), Forma de Pagamento.*

Resultado Esperado *Sistema deve informar ao usuário que o horário selecionado não está mais disponível (ex: "Este horário de agendamento acabou de se esgotar. Por favor, selecione outro."). O pedido não deve ser agendado para o horário indisponível. O sistema pode sugerir o próximo horário disponível ou retornar à tela de seleção de horários.*

Prioridade *Alta*

Pós-condições *Pedido não agendado para o horário que ficou indisponível. Itens permanecem na sacola. Usuário é informado da indisponibilidade do horário.*

6. CRONOGRAMA

Nível de Testes	#Cenários	#Roteiros	#Casos de Teste
Testes Unitários	8	9	20
Testes de Integração	5	5	8
Testes de Sistema	7	7	8
Testes de Aceitação	2	2	4
TOTAL	22	23	40

Fase	Colaborador	Data Início	Data Fim	Horas
------	-------------	-------------	----------	-------

Planejamento	Lucas / Fernando	20/05/2025	28/05/2025	4
Projeto	Lucas / Fernando	21/05/2025	28/05/2025	6
Análise	Lucas / Fernando	30/05/2025	30/05/2025	6
Execução	Lucas / Fernando	12/06/2025	30/06 /2025	16
Relato e priorização	Lucas / Fernando	30/06 /2025	30/06 /2025	4
TOTAL				36

7. CONCLUSÃO

Lucas: Ao desenvolver este plano de testes, foi possível perceber o quanto essa etapa do desenvolvimento exige atenção a cada detalhe. Durante a realização deste trabalho, ficou evidente como pequenas brechas podem resultar em grandes problemas, especialmente quando os testes não são feitos de forma adequada. Portanto, o plano de testes se mostra essencial não só para assegurar a qualidade do produto final, mas também para prevenir erros que poderiam comprometer toda a experiência do usuário.

Fernando: Desenvolver esse plano de testes foi uma experiência importante para entender na prática como cada etapa contribui para a qualidade final de um sistema. Ao pensar nos diferentes tipos de testes e criar cenários reais de uso, percebi como é essencial garantir não só que o sistema funcione, mas que ele funcione bem para o usuário. Explorar funcionalidades como login, carrinho, pedidos e pagamentos me ajudou a ver como testes bem planejados evitam falhas e melhoram a usabilidade. Mais do que seguir uma estrutura, foi uma forma de aplicar o que aprendi de forma prática e consciente, sempre pensando em entregar algo confiável e funcional