

< **TechGroceries Solutions** >

## **Descrição dos Testes do Software**

< **Shop Smart - Gerenciador de Lista de Compras** >

{ **Versão revisada < 1.2 >** }

**Autores:** **Henrico Gramacho Oliveira**  
**Paulo Roberto de Souza Carneiro Filho**  
**Lucca Dávila Bacelar**  
**Gustavo Silva Santos**  
**Elias Neves Conceição**  
**Lucas Freire Mota**  
**Fernando Augusto Melo Balbino**

< **Unifacs campus tancredo neves salvador BA** >

< **data inicial - 03/05/2024, data final - 20/05/2024** >

***Versões revisadas anteriores***

<b>Revisão</b>	<b>Comentário</b>	<b>Data</b>
1.0	Versão inicial do plano de teste	03/05/2024
1.1	Adicionados mais casos de teste	13/05/2024
1.2	Revisão geral do plano de teste	20/05/2024

# 1 Descrição dos Testes do Software

## Sumário

Versão revisadas anteriores	2
Descrição dos Testes do Software	3
Sumário	3
1 Planos de testes	4
1.1 Plano de testes de aceitação	4
1.1.1 Identificador do plano de testes	4
1.1.2 Introdução	4
1.1.3 Itens a testar	5
1.1.4 Aspectos a testar	5
1.1.5 Aspectos que não serão testados	6
1.1.6 Abordagem	6
1.1.7 Critérios de completeza e sucesso	6
1.1.8 Critérios de suspensão e retomada	6
1.1.9 Resultados dos testes	6
1.1.10 Tarefas de teste	7
1.1.11 Ambiente	7
1.1.12 Responsabilidades	7
1.1.13 Agenda	8
1.1.14 Riscos e contingências	8
1.1.15 Aprovação	8
1.2 Planos de testes de integração	8
1.2.1 Plano de testes de integração < nome da iteração >	8
2 Especificações de testes	14
2.1 Especificação do teste < nome do teste >	14
2.1.1 Identificador da especificação de teste	14
2.1.2 Aspectos a serem testados	14
2.1.3 Detalhes da abordagem	14
2.1.4 Identificação dos testes	14
2.1.5 Critérios de completeza e sucesso	15
2.1.6 Procedimentos de teste	15
2.1.7 Casos de teste	16

## 11 Planos de testes

### 11.1 Plano de testes de aceitação

#### 11.1.1 Identificador do plano de testes

**SS-GLC-TP-001**

**SS:** Abreviação do nome do projeto ou aplicativo, neste caso, "Shop Smart".

**GLC:** Abreviação de "Gerenciador de Lista de Compras", indicando o contexto específico do aplicativo.

**TP:** Abreviação de "Test Plan" (Plano de Testes).

**001:** Número sequencial, indicando que este é o primeiro plano de testes criado para este projeto.

#### 11.1.2 Introdução

##### 11.1.2.1 Objetivos dos testes

O objetivo dos testes listados é assegurar a funcionalidade, usabilidade e integração do aplicativo de gestão de lista de compras com mercados. Cada teste tem objetivos específicos que contribuem para a validação geral do sistema, garantindo que ele atenda aos requisitos de usuários e mantenha a integridade de suas funções.

##### 11.1.2.2 Histórico

Versão 1.0: Criação do plano inicial. (Data: 08/05/2024)

##### 11.1.2.3 Escopo dos testes

1. **Planejamento e Preparação:** Definir o ambiente de teste, ferramentas e dados necessários.
2. **Execução dos Testes:** Realizar os testes conforme os casos de teste definidos, registrando os resultados e os problemas encontrados.
3. **Relatórios e Análise:** Documentar os resultados dos testes, identificar falhas críticas e fornecer feedback à equipe de desenvolvimento.
4. **Revisão e Repetição:** Revisar os casos de teste com base nos resultados, corrigir problemas e repetir os testes conforme necessário.

#### 11.1.2.4 Referências a documentos relevantes

Número de ordem	Tipo do material	Referência bibliográfica
1	Documento Técnico	Documento de Requisitos do Software para o Aplicativo de Gestão de Lista de Compras
2	Documento Técnico	Especificação de Design do Sistema para o Aplicativo de Gestão de Lista de Compras
3	Documento Gerencial	Plano de Gerenciamento de Projeto para o Aplicativo de Gestão de Lista de Compras
4	Documento Técnico	Documentação da API dos Mercados Integrados
5	Documento de Referência	Manuais de Usuário do Aplicativo de Gestão de Lista de Compras
6	Documento de Normas e Regulamentos	Normas de Segurança e Privacidade para Aplicativos de Compras Online

#### 11.1.3 Itens a testar

Número de ordem	Item	Comentários
1	Funcionalidades Básicas do Aplicativo	Essas funcionalidades são essenciais para o uso básico do aplicativo. Testar a criação, edição, adição e remoção de itens de lista, bem como a marcação de itens como comprados e o compartilhamento de listas, garantirá uma experiência consistente para o usuário.
2	Integração com Mercados	A integração com mercados é uma característica central do aplicativo. Testar a sincronização de itens com informações dos mercados integrados, a atualização de preços e disponibilidade, bem como as notificações de promoções, garantirá que essa funcionalidade esteja operando conforme o esperado.
3	Usabilidade	A usabilidade é fundamental para a satisfação do usuário. Testar a interface do usuário, a navegação e a experiência geral do usuário garantirá que o aplicativo seja intuitivo e fácil de usar, aumentando a adoção e a retenção de usuários.

**11.1.4 Aspectos a testar**

Número de ordem	Item	Referência às Especificações de Testes
1	Criação de Lista de Compras	TEST-01
2	Edição de Lista de Compras	TEST-02
3	Adição de Itens à Lista de Compras	TEST-03
4	Remoção de Itens da Lista de Compras	TEST-04
5	Marcação de Itens como Comprados	TEST-05
6	Compartilhamento de Listas de Compras	TEST-06
7	Sincronização de Itens com Mercados Integrados	TEST-07
8	Atualização de Preços e Disponibilidade de Itens	TEST-08
9	Recepção de Notificações de Promoções	TEST-09
10	Efetuação de Compras	TEST-10
11	Avaliação da Interface do Usuário	TEST-11

**11.1.5 Aspectos que não serão testados**

Número de ordem	Aspecto	Motivo
1	Geração Automática de Listas de Compras	A geração automática de listas de compras com base em padrões de compra ou histórico de compras não está sendo testada neste momento devido a restrições de tempo e complexidade. Isso pode ser considerado em futuras iterações do teste.
2	Funcionalidades Avançadas de Edição de Lista	Recursos avançados de edição de lista, como categorização de itens, atribuição de prioridades ou lembretes, não são testados neste momento devido a priorização de funcionalidades básicas. Esses recursos podem ser incluídos em testes futuros conforme necessário.
3	Integração com Sistemas de Terceiros	A integração com sistemas de terceiros além dos mercados integrados não está sendo testada nesta fase. Isso inclui integração com aplicativos de pagamentos específicos ou outros aplicativos de lista de compras. A integração com esses sistemas pode ser explorada em fases de teste subsequentes.
4	Recursos de Acessibilidade	Testes detalhados de acessibilidade, como conformidade com as diretrizes de acessibilidade não estão incluídos neste momento devido a restrições de escopo e recursos. No entanto, a usabilidade básica será avaliada para garantir uma experiência acessível para o maior número possível de usuários

### 11.1.6 Abordagem

A abordagem inclui a preparação, execução, documentação e análise dos testes, bem como a gestão de qualquer risco identificado durante o processo. Esta abordagem estruturada e sistemática garantirá que todos os aspectos críticos do aplicativo sejam testados de maneira eficaz, assegurando que ele funcione conforme os requisitos e ofereça uma experiência de usuário satisfatória.

### 11.1.7 Critérios de completeza e sucesso

Número de ordem	Critério
1	Todos os casos de teste definidos no plano de testes devem ser executados. Isso inclui testes funcionais, de integração, de usabilidade, de desempenho e de segurança.
2	Todos os requisitos documentados no Documento de Requisitos do Software (SRS) e na Especificação de Design do Sistema devem ser verificados por meio de casos de teste. Nenhum requisito deve ser deixado sem teste.
3	Todos os resultados de teste devem ser documentados detalhadamente, incluindo sucesso, falhas e bugs encontrados. Cada caso de teste deve ter uma entrada correspondente no relatório de testes.
4	Todos os resultados de testes e relatórios de bugs devem ser revisados e aceitos pelos stakeholders do projeto, incluindo a equipe de desenvolvimento e os gerentes de projeto.

### 11.1.8 Critérios de suspensão e retomada

#### 11.1.8.1 Critérios de suspensão dos testes

Se uma falha crítica ou bloqueante é identificada que impede a execução de múltiplos casos de teste ou afeta funcionalidades essenciais do aplicativo (como a criação de listas de compras ou a sincronização com mercados), os testes devem ser suspensos até que a falha seja corrigida. Se o ambiente de teste (incluindo hardware, software, redes ou integrações com mercados) não está disponível ou está instável de forma que os resultados dos testes não sejam confiáveis, os testes devem ser suspensos até que o ambiente esteja estável e operacional. Se houver falta de recursos críticos para a execução dos testes, como pessoal qualificado, ferramentas de teste ou dados de teste, os testes devem ser suspensos até que os recursos necessários estejam disponíveis.

#### 11.1.8.2 Critérios de retomada dos testes

**Correção de Problemas Identificados:** A equipe de desenvolvimento deve corrigir os problemas que causaram a suspensão.

**Verificação da Solução:** Uma verificação inicial deve ser realizada para assegurar que as correções foram bem-sucedidas.

**Atualização de Casos de Teste:** Caso necessário, atualizar os casos de teste para refletir as mudanças feitas.

**Revisão e Aprovação:** A equipe de testes deve revisar e aprovar a retomada dos testes, garantindo que todas as condições para reativação foram atendidas.

**Reinício dos Testes:** Retomar os testes a partir do ponto de suspensão, seguindo o plano de testes atualizado.

### **11.1.9 Resultados dos testes**

**Correção dos Defeitos Críticos e de Alta Prioridade:** Foco na resolução dos problemas de sincronização, resiliência a erros de rede, notificações duplicadas, e atualização de informações de mercado.

**Melhoria nas Mensagens de Erro:** Tornar as mensagens de erro mais claras e orientativas para melhorar a experiência do usuário.

**Revisão e Reexecução dos Testes Falhados:** Após as correções, os casos de teste que falharam devem ser executados para garantir que os problemas foram resolvidos.

### **11.1.10 Tarefas de teste**

#### **1. Teste de Sincronização em Tempo Real:**

Verificar se a sincronização de itens com os mercados integrados ocorre em tempo real, garantindo que as informações de disponibilidade e preço sejam precisas.

#### **2. Teste de Resiliência a Erros de Rede:**

Simular condições de rede instáveis e verificar se o aplicativo é capaz de se recuperar adequadamente de erros de rede sem interromper a funcionalidade principal.

#### **3. Teste de Notificações de Promoções:**

Verificar se as notificações de promoções são enviadas corretamente aos usuários, garantindo que não sejam duplicadas e que apresentem informações precisas.

#### **4. Teste de Atualização de Informações de Mercado:**

Verificar se as atualizações de preço e disponibilidade dos itens ocorrem de forma oportuna e sem atrasos, garantindo que os usuários tenham acesso a informações atualizadas.

#### **5. Teste de Mensagens de Erro:**

Analisar e testar todas as mensagens de erro apresentadas aos usuários, garantindo que sejam claras, informativas e orientativas para ajudar os usuários a resolver problemas.

### **11.1.11 Ambiente**

#### **11.1.11.1 Hardware**

Um computador pessoal ou laptop para executar o aplicativo e realizar testes manuais. Smartphones e tablets para testar a versão móvel do aplicativo em diferentes sistemas operacionais (iOS e Android).

Uma conexão à internet estável e rápida para testar a sincronização com os mercados integrados e para simular condições de uso do mundo real.

Servidores ou ambientes de desenvolvimento e produção onde o aplicativo será implantado e integrado com os mercados.

Podem ser necessários servidores adicionais para hospedar serviços de backend e APIs.



#### 11.1.11.2 Software

**Selenium:**

Uma ferramenta de automação de testes de software para testar aplicativos da web. Pode ser usado para automatizar a interação do usuário com o aplicativo web e verificar sua funcionalidade.

**Appium:**

Uma ferramenta de automação de teste de software para aplicativos móveis. Pode ser usado para automatizar testes em dispositivos iOS e Android, simulando interações do usuário.

**Postman:**

Uma plataforma de desenvolvimento colaborativo para criar, testar e compartilhar APIs. Pode ser usado para testar a integração do aplicativo com os mercados integrados por meio de APIs.

**Cypress:** Uma ferramenta poderosa de automação de testes end-to-end, que permite testar qualquer aplicação web de forma fácil, rápida e confiável, usando a linguagem JavaScript.

#### 11.1.11.3 Ferramentas de testes

**Selenium WebDriver:**

Uma ferramenta de automação de teste amplamente utilizada para testar aplicativos da web. O Selenium WebDriver permite automatizar a interação do usuário com o aplicativo por meio de scripts, verificando a funcionalidade e a integridade do aplicativo.

**Appium:**

Uma ferramenta de automação de teste para aplicativos móveis, que suporta testes em dispositivos iOS, Android e Windows. Appium permite testar funcionalidades específicas do aplicativo em diferentes plataformas e dispositivos.

**TestComplete:**

Uma plataforma abrangente de automação de testes que suporta testes de GUI, API e testes de carga. TestComplete oferece recursos de gravação e reprodução, bem como scripting para testes mais avançados.

**Cypress:** Uma ferramenta poderosa de automação de testes end-to-end, que permite testar qualquer aplicação web de forma fácil, rápida e confiável, usando a linguagem JavaScript.

#### 11.1.11.4 Componentes de testes

##### **Casos de Teste:**

Os casos de teste são documentos que descrevem os passos a serem seguidos e os resultados esperados para validar uma funcionalidade específica do software. Cada caso de teste é projetado para testar um cenário específico e pode incluir detalhes como pré-condições, passos de teste, dados de entrada e resultados esperados.

##### **Plano de Testes:**

O plano de testes é um documento que descreve a abordagem geral para o processo de teste. Ele inclui informações sobre os objetivos dos testes, escopo, cronograma, recursos necessários e critérios de aceitação. O plano de testes fornece uma visão geral do processo de teste e ajuda a garantir que todos os aspectos críticos sejam cobertos.

##### **Estratégia de Testes:**

A estratégia de testes define a abordagem geral para a execução dos testes. Isso inclui a identificação das técnicas de teste a serem utilizadas, a alocação de recursos, a definição de métricas de qualidade e a priorização dos casos de teste.

#### 11.1.11.5 Documentos

Número de ordem	Documento necessário
1	Este documento descreve a estratégia e abordagem para testar o aplicativo de gestão de lista de compras com integração com mercados. Ele define os objetivos, escopo, cronograma e recursos necessários para o processo de teste.
2	Os objetivos dos testes são estabelecidos para garantir que o aplicativo atenda aos requisitos funcionais e não funcionais especificados. Isso inclui testar a funcionalidade básica do aplicativo, sua integração com mercados e sua usabilidade geral. Os testes visam identificar e corrigir defeitos, garantir a estabilidade e o desempenho adequado do aplicativo, e validar sua conformidade com os requisitos do usuário final.

#### 11.1.12 Responsabilidades

Número de ordem	Função	Responsabilidades
-----------------	--------	-------------------

1	Analista de Testes	O analista de testes é responsável por analisar os requisitos do software, elaborar planos de teste, criar casos de teste detalhados e executar testes funcionais e não funcionais para garantir a qualidade do software. Eles também são responsáveis por relatar e rastrear defeitos encontrados durante os testes, além de trabalhar em estreita colaboração com os desenvolvedores para resolver problemas identificados.
2	Engenheiro de Qualidade de Software	O engenheiro de qualidade de software é responsável por projetar e implementar estratégias de teste, desenvolver scripts de automação de teste, realizar testes de desempenho e segurança, e conduzir análises de qualidade de código. Eles trabalham em colaboração com equipes de desenvolvimento para garantir que os processos de teste sejam integrados ao ciclo de vida do desenvolvimento de software.
3	Líder de Testes	O líder de testes é responsável por liderar uma equipe de teste, coordenar atividades de teste, definir prioridades, alocar recursos e garantir a execução eficaz dos testes de software. Eles são responsáveis por garantir que os objetivos de teste sejam alcançados dentro do prazo e do orçamento estabelecidos, além de relatar o progresso e os resultados dos testes à gerência e outras partes interessadas.
4	Especialista em Automação de Testes	O especialista em automação de testes é responsável por desenvolver e manter frameworks de automação de teste, criar scripts de teste automatizados, e implementar estratégias de automação para aumentar a eficiência e a cobertura dos testes. Eles trabalham em estreita colaboração com a equipe de desenvolvimento para identificar oportunidades de automação e garantir a integração contínua dos testes automatizados no processo de desenvolvimento de software.

### ***11.1.13 Agenda***

Fase Inicial de Testes

**Segunda-feira:**

Configuração do ambiente de desenvolvimento e teste.

Instalação de ferramentas de teste.

Configuração de dispositivos de teste.

**Terça-feira:**

Configuração das conexões com as APIs dos mercados.

Importação de dados de teste iniciais.

**Quarta-feira:**

Verificação da configuração do ambiente.

Revisão das especificações dos casos de teste.

**Quinta-feira:**

Treinamento da equipe de teste sobre o uso do aplicativo e ferramentas.

Discussão e revisão dos procedimentos de teste.

**Sexta-feira:**

Reunião de planejamento para a execução dos casos de teste.

Ajustes finais no ambiente de teste.

### 11.1.14 Riscos e contingências

Número	Risco	Gravidade	Probabilidade de ocorrência	Impacto previsto	Contramedidas previstas
1	Falha na Integração Com Mercados	Alta (A)	Média (M)	Problemas na comunicação entre o aplicativo e os sistemas dos mercados parceiros podem impedir a atualização das listas de preços e estoque em tempo real.	Estabelecer ambientes de teste separados para integração e implementar mocks para simular respostas dos sistemas de mercado. Além disso, desenvolver procedimentos de fallback que permitam o uso de dados de cache quando a integração falhar.
2	Problemas de Desempenho Sob Alta Carga	Alta (A)	Média (M)	O aplicativo pode não funcionar corretamente sob alta carga de usuários simultâneos, levando a lentidão ou falhas.	Realizar testes de carga e estresse para identificar e corrigir gargalos de desempenho. Implementar mecanismos de balanceamento de carga e escalabilidade para lidar com picos de tráfego.
3	Defeitos Críticos Não Identificados em Ambientes de Teste	Alta (A)	Média (M)	Alguns defeitos críticos podem não ser detectados em ambientes de teste devido à falta de dados reais ou variações nos cenários de uso.	Implementar testes beta com um grupo selecionado de usuários reais para identificar problemas que não aparecem em ambientes de teste. Coletar e analisar feedback para corrigir defeitos antes do lançamento geral.
4	Problemas de Usabilidade	Média (M)	Alta (A)	O aplicativo pode ser difícil de usar ou não intuitivo, o que pode levar à insatisfação dos usuários e baixo engajamento.	Conduzir testes de usabilidade com usuários finais para identificar problemas e áreas de melhoria. Iterar no design da interface com base no feedback recebido e realizar treinamentos com a equipe de suporte ao cliente para ajudar os usuários a superar dificuldades.

### 11.1.15 Aprovação

Aprovamos o documento do Plano de Testes de Aceitação do projeto < **Shop Smart** >

Nome	Organização	Data	Assinatura
Henrico Gramacho Oliveira	Gerente de Projeto	13/05/2024	12722130475
Paulo Roberto Souza Carneiro Filho	Líder de Testes	13/05/2024	1272215237
Lucca Dávila Bacelar	Desenvolvedor Principal	13/05/2024	1272218370
Gustavo Silva Santos	Analista de Qualidade	13/05/2024	1272214938
Elias Neves Conceição	Representante do Cliente	13/05/2024	12722127604
Lucas Freire Mota	Representante do Cliente	13/05/2024	12722217127
Fernando Augusto Melo Balbino	Desenvolvedor Principal	13/05/2024	12723130856

## 11.2 Planos de testes de integração

### 11.2.1 Plano de testes de integração < PTI-001 >

#### 11.2.1.1 Identificador do plano de testes

(PTI-001) Aplicativo de Gestão de Lista de Compras com Integração com Mercados

SS-GLC-TP-002

**SS:** Abreviação do nome do projeto ou aplicativo, neste caso, "Shop Smart".

**GLC:** Abreviação de "Gerenciador de Lista de Compras", indicando o contexto específico do aplicativo.

**TP:** Abreviação de "Test Plan" (Plano de Testes).

**002:** Número sequencial, indicando que este é o primeiro plano de testes criado para este projeto.

### 11.2.1.2 Introdução

#### 11.2.1.2.1 Objetivos dos testes

Verificar a correta integração entre o aplicativo de gestão de lista de compras e os sistemas dos mercados parceiros. Garantir a atualização em tempo real de preços e estoque. Assegurar a interoperabilidade entre diferentes sistemas e componentes do aplicativo.

#### 11.2.1.2.2 Histórico

Versão 2.0: Criação do plano inicial. (Data: 21/05/2024)

#### 11.2.1.2.3 Escopo dos testes

**Inclusões:** Testes de comunicação com APIs de mercados, sincronização de dados (preços e estoque), autenticação e autorização, e testes de fluxo de dados entre o aplicativo e os mercados.  
**Exclusões:** Testes de interface do usuário (UI) que não envolvem integração, testes de usabilidade e testes de desempenho não relacionados à integração.

#### 11.2.1.2.4 Referências a documentos relevantes

Número de ordem	Tipo do material	Referência bibliográfica
1	Documento de Requisitos de Sistema	Documento de Requisitos Funcionais e Não Funcionais.
2	Especificação Técnica de Integração	Detalhamento técnico das APIs dos mercados parceiros.
3	Guia de Testes de Integração	Procedimentos e boas práticas para testes de integração.
4	Plano de Gerenciamento de Projetos	Plano de projeto com cronograma e alocação de recursos.
5	Relatório de Testes de Desempenho	Resultados dos testes de carga e estresse.
6	Política de Segurança da Informação	Normas e diretrizes para proteção de dados.

#### 11.2.1.3 Itens a testar

Número de ordem	Item	Comentários
1	Comunicação com APIs dos Mercados:	Verificar se as chamadas às APIs dos mercados são bem-sucedidas e retornam os dados corretos.
2	Sincronização de Dados:	Testar a atualização em tempo real de preços e estoque no aplicativo.
3	Autenticação e Autorização:	Garantir que o processo de autenticação e autorização para acesso aos dados dos mercados esteja funcionando corretamente.

#### 11.2.1.4 Aspectos a testar

Número de ordem	Item	Referência às Especificações de Testes
1	Validação de Conexão com APIs (AT-001)	<b>Especificação:</b> Testar a conexão inicial com as APIs dos mercados.
2	Sincronização de Preços	<b>Especificação:</b> Verificar se os preços dos produtos são atualizados corretamente.
3	Sincronização de Estoque	<b>Especificação:</b> Testar a atualização dos níveis de estoque em tempo real.
4	Manuseio de Erros de API	<b>Especificação:</b> Validar o tratamento de erros e respostas inesperadas das APIs.
5	Autenticação de Usuário	<b>Especificação:</b> Testar o processo de login com credenciais de mercado.
6	Autorização de Acesso	<b>Especificação:</b> Verificar se apenas usuários autorizados podem acessar dados específicos.
7	Desempenho da Sincronização	<b>Especificação:</b> Testar a rapidez e eficiência da sincronização de dados.



8	Segurança na Transmissão de Dados	<b>Especificação:</b> Garantir que a transmissão de dados entre o aplicativo e os mercados é segura.
9	Integração de Dados de Cache	<b>Especificação:</b> Testar o uso de dados de cache em caso de falha na integração.
10	Testes de Regressão Pós-Atualização	<b>Especificação:</b> Garantir que novas atualizações não afetem a integração existente.
11	Testes de Backup e Recuperação	<b>Especificação:</b> Verificar se os dados podem ser recuperados em caso de falha.

#### 11.2.1.5 Aspectos que não serão testados

Número de ordem	Aspecto	Motivo
1	Testes de Interface do Usuário (UI) não relacionados à integração	<b>Motivo:</b> Estes testes serão cobertos em um plano de testes separado.
2	Testes de Usabilidade	<b>Motivo:</b> Foco principal nos testes de integração técnica.
3	Testes de Documentação	<b>Motivo:</b> Não afeta diretamente a integração técnica.
4	Testes de Compatibilidade de Dispositivos	<b>Motivo:</b> Não diretamente relacionados à funcionalidade de integração.

#### 11.2.1.6 Abordagem

**Metodologia:** Testes manuais e automatizados.

**Ferramentas:** Postman para testes de API, Selenium para testes automatizados, JMeter para testes de desempenho.

**Ambiente de Teste:** Ambiente de desenvolvimento e testes dedicado, com acesso às APIs dos mercados.

#### 11.2.1.7 Critérios de completudeza e sucesso

Número de ordem	Critério
1	Todas as conexões com as APIs dos mercados devem ser bem-sucedidas.
2	Todos os dados de preços e estoque devem ser sincronizados em tempo real.
3	Nenhum erro crítico deve ser encontrado durante a fase de testes.
4	O tempo de resposta das sincronizações deve estar dentro dos limites aceitáveis.

#### 11.2.1.8 Critérios de suspensão e retomada

##### 11.2.1.8.1 Critérios de suspensão dos testes

**Falhas Graves:** Testes serão suspensos se houver falhas críticas que impossibilitem a continuação dos testes.

**Problemas de Ambiente:** Testes serão suspensos se houver problemas no ambiente de teste que comprometam a validade dos resultados.

**Falta de Recursos:** Testes serão suspensos se não houver recursos necessários (humanos ou técnicos) para continuar.

**Problemas de Integração:** Testes serão suspensos se houver falhas recorrentes e não resolvidas na integração.

##### 11.2.1.8.2 Critérios de retomada dos testes

**Critério:** Todos os defeitos críticos identificados durante a fase de testes anteriores devem ser corrigidos.

**Verificação:** Revisão e revalidação dos defeitos corrigidos devem ser realizadas para garantir que os problemas foram resolvidos adequadamente.

**Responsável:** Equipe de desenvolvimento, com validação pela equipe de testes.

#### 11.2.1.9 Resultados dos testes

Um relatório detalhado será gerado ao final dos testes, documentando os resultados, defeitos encontrados, e recomendações para correções e melhorias.

#### 11.2.1.10 Tarefas de teste

Criação de Casos de Teste  
Configuração do Ambiente de Teste  
Execução de Casos de Teste  
Relatório de Defeitos  
Execução de Testes após Correções

#### 11.2.1.11 Ambiente

##### 11.2.1.11.1 Hardware

**Servidores de Teste:** Configurados para simular o ambiente de produção.  
**Dispositivos Móveis:** Vários dispositivos móveis para testes de compatibilidade.

##### 11.2.1.11.2 Software

**Sistema Operacional:** Windows/Linux para servidores, Android/iOS para dispositivos móveis.  
**Ferramentas de Teste:** Postman, Selenium, JMeter.

##### 11.2.1.11.3 Ferramentas de testes

**Postman:** Para testes de API.  
**Selenium:** Para testes automatizados de interface.  
**JMeter:** Para testes de carga e desempenho.

##### 11.2.1.11.4 Componentes de testes

**APIs dos Mercados:** Componentes para comunicação e sincronização de dados.  
**Base de Dados:** Para armazenamento e recuperação de dados sincronizados.  
**Módulos de Autenticação e Autorização:** Para garantir segurança no acesso aos dados.

##### 11.2.1.11.5 Documentos

Número de ordem	Documento necessário
1	Plano de Testes de Integração (PTI-001): Este documento.
2	Relatório de Resultados de Testes (RRT-001): Documento detalhando os resultados dos testes.

#### 11.2.1.12 Responsabilidades

Número de ordem	Função	Responsabilidades
1	Analista de Testes	Criar e executar casos de teste.
2	Engenheiro de Qualidade	Desenvolver scripts de automação e garantir a cobertura de testes.
3	Líder de Testes	Coordenar a equipe e gerenciar o plano de testes.
4	Especialista em Integração	Garantir a comunicação correta entre os sistemas.

#### 11.2.1.13 Agenda

Fase de Testes de Integração e Reteste

**Segunda-feira:**

Testes de integração com os sistemas de pagamento dos mercados.

Registro de resultados e documentação de falhas.

**Terça-feira:**

Testes de integração com o sistema de inventário dos mercados.

Registro de resultados e documentação de falhas.

**Quarta-feira:**

Testes de integração com a funcionalidade de promoções e descontos dos mercados.

Registro de resultados e documentação de falhas.

**Quinta-feira:**

Testes de integração com o sistema de envio de notificações.

Registro de resultados e documentação de falhas.

**Sexta-feira:**

Revisão e análise dos resultados do teste da semana.

Relatório de progresso e identificação de defeitos críticos.

## 11.2.1.14

Número	Risco	Gravidade	Probabilidade de ocorrência	Impacto previsto	Contramedidas previstas
1	Inconsistências de Dados entre Aplicativo e Mercados	Alta (A)	Média (M)	Pode haver inconsistências entre os dados exibidos no aplicativo e os dados reais dos mercados (preços, estoque, etc.).	Implementar verificações de consistência periódicas entre os dados do aplicativo e os dados dos mercados. Utilizar logs detalhados para identificar e corrigir rapidamente quaisquer discrepâncias. Notificar os usuários sobre possíveis inconsistências e fornecer uma atualização assim que os dados corretos forem sincronizados.
2	Falhas de Segurança nas Transmissões de Dados	Alta (A)	Média (M)	Dados sensíveis podem ser interceptados durante a transmissão entre o aplicativo e os mercados, levando a brechas de segurança.	Usar criptografia SSL/TLS para todas as transmissões de dados. Realizar auditorias de segurança regulares e testes de penetração para identificar e corrigir vulnerabilidades. Implementar autenticação multifator (MFA) para usuários acessando informações críticas.
3	Problemas de Conectividade de Rede	Alta (A)	Média (M)	Problemas de conectividade de rede podem impedir que o aplicativo se comunique com os servidores dos mercados.	Implementar uma lógica de reconexão automática no aplicativo. Oferecer uma funcionalidade offline onde os dados mais recentes são armazenados localmente e sincronizados quando a conexão for restabelecida. Utilizar serviços de monitoramento de rede para detectar e resolver problemas de conectividade rapidamente.
4	Dependência de Terceiros (APIs dos Mercados)	Média (M)	Alta (A)	O funcionamento do aplicativo depende de APIs de terceiros, que podem ser alteradas ou desativadas sem aviso prévio.	Estabelecer acordos de nível de serviço (SLAs) com os provedores das APIs e manter uma comunicação ativa para antecipar mudanças. Implementar um sistema de fallback que utilize APIs alternativas ou dados locais em caso de falha das APIs principais. Monitorar continuamente o desempenho e a disponibilidade das APIs de terceiros.



### ***11.2.2.1 Aprovação***

Aprovamos o documento do Plano de Testes de Aceitação do projeto < **Shop Smart** >

<b>Nome</b>	<b>Organização</b>	<b>Data</b>	<b>Assinatura</b>
Henrico Gramacho Oliveira	Gerente de Projeto	20/05/2024	12722130475
Paulo Roberto Souza Carneiro Filho	Líder de Testes	20/05/2024	1272215237
Lucca Dávila Bacelar	Desenvolvedor Principal	20/05/2024	1272218370
Gustavo Silva Santos	Analista de Qualidade	20/05/2024	1272214938
Elias Neves Conceição	Representante do Cliente	20/05/2024	12722127604
Lucas Freire Mota	Representante do Cliente	20/05/2024	12722217127
Fernando Augusto Melo Balbino	Desenvolvedor Principal	20/05/2024	12723130856





## 12 Especificações de testes

### 12.1 Especificação do teste < **ET-001** >

#### 12.1.1 Identificador da especificação de teste

(ET-001) Aplicativo de Gestão de Lista de Compras com Integração com Mercados

SS-GLC-TP-003

**SS:** Abreviação do nome do projeto ou aplicativo, neste caso, "ShopSmart".

**GLC:** Abreviação de "Gerenciador ", indicando o contexto específico do aplicativo.

**TP:** Abreviação de "Test Plan" (Plano de Testes).

**003:** Número sequencial, indicando que este é o primeiro plano de testes criado para este projeto.

#### 12.1.2 Aspectos a serem testados

Número	Requisito	Comentários
1	Usabilidade da Interface de Usuário	Avaliar a facilidade de uso e a experiência do usuário com a interface do aplicativo. Realizar testes de usabilidade com um grupo de usuários representativos.
2	Compatibilidade com Diferentes Dispositivos	Verificar o funcionamento do aplicativo em vários dispositivos e sistemas operacionais (iOS, Android). Testar o aplicativo em uma variedade de dispositivos móveis e tablets.
3	Notificações e Alertas	Testar a funcionalidade de notificações push e alertas no aplicativo. Configurar e verificar a recepção de notificações em diferentes cenários de uso.

### 12.1.3 Detalhes da abordagem

**Testes Funcionais:** Validar se todas as funcionalidades do aplicativo estão operando conforme o esperado.

**Testes de Integração:** Garantir que os diferentes módulos do aplicativo interajam corretamente entre si e com as APIs dos mercados.

### 12.1.4 Identificação dos testes

#### 12.1.4.1 Procedimentos de teste

Número	Procedimento de teste	Identificação do procedimento de teste
1	Preparação do Ambiente de Teste	Configurar e preparar o ambiente necessário para a execução dos testes.
2	Execução dos Casos de Teste	Executar os casos de teste conforme planejado para validar as funcionalidades e desempenho do aplicativo.
3	Revisão e Análise dos Resultados de Teste	Analisar os resultados dos testes, identificar problemas e determinar as ações corretivas necessárias.

#### 12.1.4.2 Casos de teste

Número	Caso de teste	Identificação do caso de teste
1	Verificar se um novo usuário pode se cadastrar com sucesso.	Navegar para a tela de cadastro, preencher os campos obrigatórios, e submeter o formulário.
2	Verificar se um usuário existente pode fazer login.	Navegar para a tela de login, inserir credenciais válidas, e submeter o formulário.
3	Verificar se um usuário pode adicionar um item à sua lista de compras.	Navegar para a tela da lista de compras, adicionar um item preenchendo os detalhes necessários.
4	Verificar se um usuário pode editar um item existente na lista de compras	Selecionar um item da lista de compras, alterar os detalhes e salvar
5	Verificar se um usuário pode remover um item da lista de compras.	Selecionar um item e escolher a opção de remoção.
6	Verificar se os dados da lista de compras são sincronizados corretamente com os dados do mercado (preços e estoque).	Adicionar itens à lista de compras e verificar a atualização dos dados de preço e estoque.
7	Verificar se o usuário pode pesquisar produtos disponíveis no mercado através do aplicativo.	Usar a funcionalidade de busca para procurar por um produto específico.
8	Verificar se o usuário recebe notificações de promoções e ofertas.	Simular uma promoção no sistema do mercado e verificar a notificação no aplicativo.
9	Verificar o desempenho do aplicativo com múltiplos usuários simultâneos.	Simular múltiplos usuários interagindo com o aplicativo ao mesmo tempo.
10	Avaliar a facilidade de uso do aplicativo.	Realizar uma sessão de teste de usabilidade com usuários representativos, observando sua interação com o aplicativo.

#### 12.1.5 Critérios de completeza e sucesso

Número	Critério
1	Cobertura de Teste <b>Critério:</b> Todos os casos de teste planejados foram executados.
2	Taxa de Passagem <b>Critério:</b> Pelo menos 95% dos casos de teste passaram com sucesso.
3	Correção de Defeitos Críticos <b>Critério:</b> Todos os defeitos críticos e de alta prioridade foram corrigidos e avaliados.

### 12.1.6 Procedimentos de teste

#### 12.1.6.1 Procedimento de teste < Autenticação de Usuário >

<b>Identificação</b>	PT-001
<b>Objetivo</b>	O objetivo deste procedimento de teste é verificar se o sistema de autenticação de usuário do aplicativo de gestão de lista de compras com integração com mercados funciona corretamente, permitindo que os usuários acessem o sistema com suas credenciais de forma segura e eficaz.
<b>Requisitos especiais</b>	O sistema deve fornecer um formulário de login na tela inicial do aplicativo. Os usuários devem ser capazes de inserir seu nome de usuário e senha para acessar o aplicativo. O sistema deve validar as credenciais do usuário e permitir o acesso apenas a usuários autorizados. Mensagens de erro apropriadas devem ser exibidas em caso de falha na autenticação.
<b>Fluxo</b>	<p><b>Acesso à Tela de Login:</b></p> <ul style="list-style-type: none"> <li>• Abrir o aplicativo e acessar a tela inicial.</li> <li>• Verificar se há um formulário de login visível na tela.</li> </ul> <p><b>Inserção de Credenciais Inválidas:</b></p> <ul style="list-style-type: none"> <li>• Inserir um nome de usuário inválido e uma senha inválida nos campos de login.</li> <li>• Clicar no botão de login e verificar se uma mensagem de erro apropriada é exibida indicando que as credenciais são inválidas.</li> </ul> <p><b>Inserção de Credenciais Válidas:</b></p> <ul style="list-style-type: none"> <li>• Inserir um nome de usuário válido e uma senha válida nos campos de login.</li> <li>• Clicar no botão de login e verificar se o acesso é concedido ao sistema.</li> </ul> <p><b>Verificação de Acesso Negado:</b></p> <ul style="list-style-type: none"> <li>• Tentar acessar o sistema com credenciais inválidas.</li> <li>• Verificar se o acesso é negado e uma mensagem de erro apropriada é exibida.</li> </ul> <p><b>Verificação de Acesso Autorizado:</b></p> <ul style="list-style-type: none"> <li>• Acessar o sistema com credenciais válidas.</li> <li>• Verificar se o acesso é concedido e o usuário é redirecionado para a página inicial do aplicativo.</li> </ul> <p><b>Teste de Recuperação de Senha:</b></p> <ul style="list-style-type: none"> <li>• Testar o processo de recuperação de senha, inserindo um endereço de e-mail válido.</li> <li>• Verificar se um e-mail de recuperação de senha é enviado com sucesso ao usuário.</li> </ul> <p><b>Verificação de Segurança:</b></p> <ul style="list-style-type: none"> <li>• Tentar acessar o sistema sem inserir nenhuma credencial.</li> <li>• Verificar se o acesso é negado e uma mensagem de erro é exibida, indicando que as credenciais são obrigatórias.</li> </ul> <p><b>Teste de Sessão Ativa:</b></p> <ul style="list-style-type: none"> <li>• Após fazer login, fechar o aplicativo e reabri-lo.</li> <li>• Verificar se o usuário permanecer conectado e não precisa fazer login novamente.</li> </ul> <p><b>Teste de Sessão Expirada:</b></p> <ul style="list-style-type: none"> <li>• Ficar inativo por um período prolongado após fazer login.</li> <li>• Verificar se a sessão expira automaticamente e o usuário é redirecionado para a tela de login ao tentar realizar uma ação.</li> </ul> <p><b>Teste de Logoff:</b></p> <ul style="list-style-type: none"> <li>• Realizar o logoff manualmente após fazer login.</li> <li>• Verificar se o usuário é desconectado com sucesso e redirecionado para a tela de login.</li> </ul>

**12.1.7 Casos de teste****12.1.7.1 Caso de teste < Login com Credenciais Válidas >**

<b>Identificação</b>	LCV-001	
<b>Itens a testar</b>	Verificar se o sistema permite o acesso com credenciais válidas. Verificar se o usuário é redirecionado para a página inicial após o login bem-sucedido.	
<b>Entradas</b>	<b>Campo</b>	<b>Valor</b>
	Nome de usuario: "usuario_teste"	As credenciais fornecidas (nome de usuário e senha) são válidas e correspondem a um usuário registrado no sistema.
	Senha: "senha123"	O dispositivo utilizado para o login é um smartphone Android, IOS ou um desktop
<b>Saídas esperadas</b>	<b>Campo</b>	<b>Valor</b>
	O sistema deve permitir o acesso ao usuário com as credenciais fornecidas.	O acesso é concedido ao usuário, indicando que as credenciais são válidas e o login foi bem-sucedido.
	O usuário deve ser redirecionado para a página inicial do aplicativo após o login bem-sucedido.	O usuário é redirecionado para a página inicial do aplicativo após o login.
<b>Ambiente</b>	Dispositivo móvel ou emulador com o aplicativo instalado. Conexão de internet estável.	
<b>Procedimento</b>	<ol style="list-style-type: none"> <li>1. Abrir o aplicativo de gestão de listas de compras.</li> <li>2. Na tela de login, inserir o nome de usuário "usuario_teste" no campo de nome de usuário.</li> <li>3. Inserir a senha "senha123" no campo de senha.</li> <li>4. Clicar no botão de login.</li> <li>5. Verificar se o acesso é concedido e o usuário é redirecionado para a página inicial do aplicativo.</li> </ol>	
<b>Dependências</b>	Os serviços de autenticação e backend do aplicativo devem estar operacionais. As credenciais de usuário fornecidas devem estar registradas no sistema e ser válidas.	