

Plano de Testes para a API ServeRest

1. Objetivo

O objetivo principal deste plano de testes é garantir a qualidade e a conformidade da API ServeRest com os requisitos funcionais e não funcionais, conforme especificado nas histórias de usuário e no mapa mental de estratégias de teste fornecidos. Busca-se identificar defeitos, reduzir riscos e validar que a API atende às expectativas de desempenho, segurança e usabilidade.

4. Escopo

O escopo deste plano de testes abrange a API ServeRest, versão 2.29.7, hospedada localmente usando NPM e NODE.JS em `https://localhost:3000`, incluindo todos os seus módulos e endpoints, conforme detalhado nos Itens a Serem Testados e Funcionalidades a Serem Testadas.

4.1. Itens a Serem Testados

- Módulo de Usuários: POST /usuarios, GET /usuarios, GET /usuarios/{_id}, PUT /usuarios/{_id}, DELETE /usuarios/{_id}.
- Módulo de Login: POST /login.
- Módulo de Produtos: POST /produtos, GET /produtos, GET /produtos/{_id}, PUT /produtos/{_id}, DELETE /produtos/{_id}.
- Módulo de Carrinhos: GET /carrinhos, POST /carrinhos, GET /carrinhos/{_id}, DELETE /carrinhos/concluir-compra, DELETE /carrinhos/cancelar-compra.

4.2. Funcionalidades a Serem Testadas

As funcionalidades a serem testadas são derivadas das histórias de usuário fornecidas e da exploração da API:

- US 001 - Gerenciamento de Usuários: Cadastro, listagem, busca, atualização e exclusão de usuários, incluindo validações de campos, unicidade de e-mail, restrições de provedores e regras de senha.
- US 002 - Autenticação de Usuários (Login): Autenticação de usuários válidos e inválidos, geração e validação de token Bearer, e controle de acesso a rotas protegidas.
- US 003 - Gerenciamento de Produtos: Cadastro, listagem, busca, atualização e exclusão de produtos, incluindo validações de nome único e restrições de exclusão.

- Módulo de Carrinhos (Funcionalidade Sem User Stories!): Listagem, cadastro, busca, conclusão e cancelamento de compras, com foco na interação com produtos.

4.3. Funcionalidades Não a Serem Testadas

- Interface de Usuário (Frontend): O foco é exclusivamente na API (backend).
- Integrações com Sistemas Externos (não especificados): Apenas as interações internas da API ServeRest serão testadas.
- Testes de Usabilidade da Interface: Não aplicável ao teste de API.

5. Análise

Esta análise foi fundamentada em:

- Histórias de Usuário: Detalhamento das expectativas dos usuários para as funcionalidades de Usuários, Login e Produtos, incluindo critérios de aceitação específicos.
- Mapa Mental de Estratégias de Teste: Forneceu insights sobre abordagens, técnicas, tipos de teste, ferramentas, massas de dados, notas e riscos, servindo como base para a estratégia de teste.
- Documentação da API (Swagger/OpenAPI): Exploração dos endpoints, métodos, parâmetros e modelos de dados da API, permitindo a identificação de funcionalidades explícitas e implícitas, como o módulo de Carrinhos.
- Padrões e Normas: Análise das diretrizes do ISTQB CTFL 4.0, ISO/IEC 25010:2011, ISO/IEC/IEEE 29119 e IEEE 829 para garantir a conformidade e a abrangência do plano de testes.

Esta análise permitiu a identificação de cenários de teste, a definição de critérios de aceitação e a categorização dos testes em funcionais e não funcionais, além de subsidiar a matriz de risco e a estratégia de automação.

6. Técnicas Aplicadas

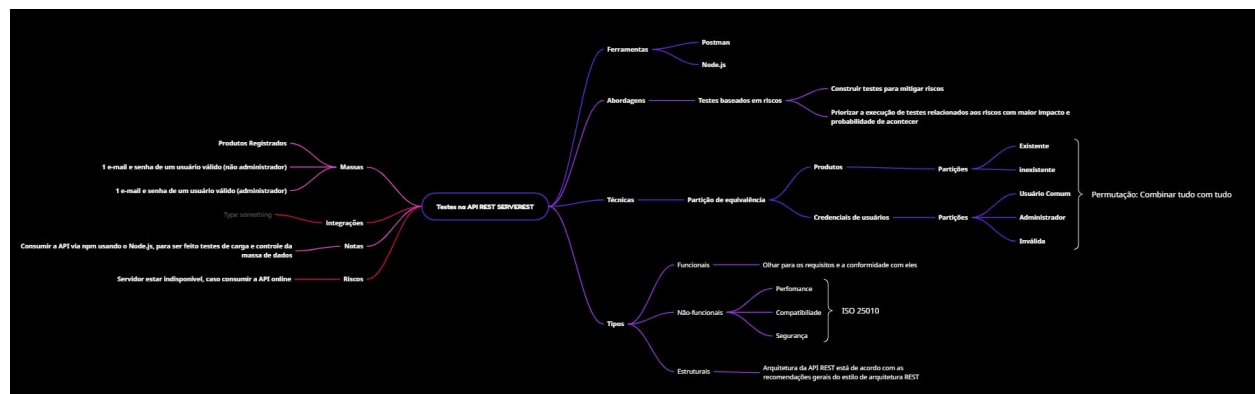
A abordagem de teste será multifacetada, combinando diferentes estratégias e técnicas para garantir uma cobertura abrangente e eficaz. Os princípios de teste do ISTQB CTFL 4.0 guiarão todas as atividades de teste.

6.1. Estratégias e Técnicas de Design de Teste

- Testes Baseados em Riscos (Risk-Based Testing): Priorização dos testes com base na identificação e avaliação dos riscos, focando em falhas críticas em endpoints, vulnerabilidades de autenticação e segurança. Isso se alinha com o princípio de que o teste é dependente do contexto e que o teste exaustivo é impossível.

- Testes Shift-Left: Início das atividades de teste o mais cedo possível, incluindo a validação da documentação da API (Swagger/OpenAPI) antes da implementação. Isso reforça o princípio do teste antecipado.
- Testes Exploratórios e Baseados em Sessão: Complementar a cobertura dos testes automatizados com a descoberta de defeitos não previstos, alinhado com o princípio de que o teste mostra a presença de defeitos e o paradoxo do pesticida.
- Técnicas de Design de Teste:
 - Particionamento de Equivalência (EP): Para otimizar a seleção de dados de teste, dividindo entradas em classes válidas e inválidas.
 - Análise de Valor Limite (BVA): Focando nos limites das partições para identificar defeitos onde eles são mais prováveis.
 - Teste de Decisão: Para testar regras de negócio complexas e condições lógicas.
 - Permutação e Cobertura Combinatória: Para testar combinações de entradas e condições, abordando a complexidade de interações.

7. Mapa Mental da Aplicação



8. Cenários de Teste Planejados

Esta seção detalha os cenários e casos de teste para as funcionalidades da API ServeRest, incorporando as características de qualidade da ISO/IEC 25010:2011 e as técnicas de design de teste. Cada caso de teste inclui um ID único, descrição, pré-condições, dados de entrada, resultado esperado, observações, tipo de teste, prioridade, característica de qualidade e técnica de design.

8.1. Cenários e Casos de Teste para US 001 - Usuários

Esta seção detalha os cenários e casos de teste para a funcionalidade de gerenciamento de usuários, incorporando as características de qualidade da ISO/IEC 25010:2011 e as técnicas de design de teste.

8.1.1. Cenário: Cadastro de Novo Usuário

Objetivo: Validar o cadastro bem-sucedido de novos usuários com dados válidos e a rejeição de cadastros com dados inválidos ou duplicados, garantindo a Adequação Funcional (Completeness, Correctness, Appropriateness) e Segurança (Confidentiality, Integrity).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US001-CT001	Cadastrar usuário com todos os campos válidos (usuário comum)	N/A	Nome: Usuário Teste, Email: usuario.teste@example.com, Password: senha123, Administrator: false	Status 201 Created, usuário cadastrado com sucesso.		Funcional	Alta	Adequação Funcional	Participação de Equivalência
US001-CT002	Cadastrar usuário com todos os campos válidos	N/A	Nome: Admin Teste, Email: admin.teste@example.com	Status 201 Created, usuário administrador cadastrado		Funcional	Alta	Adequação Funcional	Participação de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	(administrador)		com, Password: admin123, Administrador: true	rado com sucesso.					
US001-CT003	Tentar cadastrar usuário com e-mail já utilizado	Usuário usuario.existe@example.com já cadastrado	Nome: Novo Usuário, Email: usuario.teste@example.com, Password: nova123, Administrador: false	Status 400 Bad Request, mensagem de erro indicando e-mail já utilizado.		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência
US001-CT004	Tentar cadastrar usuário com e-mail inválido	N/A	Nome: Usuário Inválido, Email: email invalido	Status 400 Bad Request, mensagem de erro		Funcional	Média	Adequação Funcional (Corretude)	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	o (sem @)		ido.com, Password: senha 123, Administrador: false	de formato de e-mail.					
US001-CT005	Tentar cadastrar usuário com e-mail inválido (sem domínio)	N/A	Nome: Usuário Inválido, Email: email invalido@, Password: senha 123, Administrador: false	Status 400 Bad Request, mensagem de erro de formato de e-mail.		Funcional	Média	Adequação Funcional (Corretude)	Participante de Equivalência
US001-CT006	Tentar cadastrar usuário com e-mail de proved	N/A	Nome: Usuário Gmail, Email: teste@gmail	Status 400 Bad Request, mensagem de erro		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	or gmail.com		l.com, Password: senha 123, Administrador: false	indicando provedor não permitido.					
US001-CT007	Tentar cadastrar usuário com e-mail de provedor hotmail.com	N/A	Nome: Usuário Hotmail, Email: teste@hotmail.com, Password: senha 123, Administrador: false	Status 400 Bad Request, mensagem de erro indicando provedor não permitido.		Funcional	Alta	Adequação Funcional (Corretude)	Particionamento de Equivalência
US001-CT008	Tentar cadastrar usuário com senha menor	N/A	Nome: Usuário Senha Curta, Email:	Status 400 Bad Request, mensagem		Funcional	Média	Adequação Funcional (Corretude)	Análise de Valor Limite

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	que 5 caracteres		curta @example.com, Password: 1234, Administrator: false	de erro indicando senha muito curta.					
US001-CT009	Tentar cadastrar usuário com senha maior que 10 caracteres	N/A	Nome: Usuário Senha Longa, Email: longa@example.com, Password: 12345678901, Administrator: false	Status 400 Bad Request, mensagem de erro indicando senha muito longa.		Funcional	Média	Adequação Funcional (Corretude)	Análise de Valor Limite
US001-CT010	Tentar cadastrar usuário	N/A	Nome: "", Email: vazio	Status 400 Bad Request		Funcional	Média	Adequação Funcional	Participante de

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	o com campo nome vazio		@example.com, Password: `senha123`, Administrador: `false`	st, mensagem de erro indicando campo nome obrigatório.				(Completeness)	Equivalence
US001-CT011	Tentar cadastrar usuário com campo email vazio	N/A	Nome: Usuário Vazio, Email: ``, Password: senha123, Administrador: `false`	Status 400 Bad Request, mensagem de erro indicando campo email obrigatório/inválido.		Funcional	Média	Adequação Funcional (Completeness)	Particionamento de Equivalence
US001-CT012	Tentar cadastrar usuário com campo password vazio	N/A	Nome: Usuário Vazio, Email: vazio2@example.	Status 400 Bad Request, mensagem de erro indicando		Funcional	Média	Adequação Funcional (Completeness)	Particionamento de Equivalence

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			com, Password: ``, Administrador: false	do campo password obrigatório/inválido.					
US001-CT013	Tentar cadastrar usuário com campo administrador inválido (não booleano)	N/A	Nome: Usuário Inválido, Email: invalido@example.com, Password: senha 123, Administrador: string	Status 400 Bad Request, mensagem de erro indicando o tipo de dado inválido para administrador.		Funcional	Média	Adequação Funcional (Corretude)	Participante de Equivalência

8.1.2. Cenário: Listagem e Busca de Usuários

Objetivo: Validar a correta listagem e busca de usuários cadastrados, garantindo a Adequação Funcional (Completeness, Correctness) e Eficiência de Desempenho (Performance).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US001-CT014	Listar todos os usuários cadastrados	Vários usuários cadastrados	N/A	Status 200 OK, lista de usuários contendo todos os usuários cadastrados.		Funcional	Alta	Adequação Funcional	Participante de Equivalência
US001-CT015	Listar usuários quando não há nenhum usuário cadastrado	Nenhum usuário cadastrado	N/A	Status 200 OK, lista de usuários vazia.		Funcional	Média	Adequação Funcional	Participante de Equivalência
US001-CT016	Buscar usuário por ID válido	Usuário com ID XYZ cadastrado	ID: XYZ	Status 200 OK, dados do usuário XYZ retornados.		Funcional	Alta	Adequação Funcional	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US001-CT017	Buscar usuário por ID inexistente	N/A	ID: ABC (inexistente)	Status 400 Bad Request (ou 404 Not Found, dependendo da implementação), mensagem de erro indicando usuário não encontrado.		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência
US001-CT018	Buscar usuário por ID com formato inválido	N/A	ID: 123-abc (formato inválido)	Status 400 Bad Request, mensagem de erro indicando formato de		Funcional	Média	Adequação Funcional (Corretude)	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
				ID inválido.					

8.1.3 Cenário: Atualização de Usuário

Objetivo: Validar a atualização de dados de usuários existentes e a criação de novos usuários via PUT quando o ID não existe, garantindo a Adequação Funcional (Corretude, Completude) e Segurança (Integridade).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US001-CT019	Atualizar dados de usuário existente com sucesso	Usuário com ID XYZ cadastrado	ID: XYZ, Nome: Usuário Atualizado, Email: atualizado@example.com, Password: novaSenha, Administrador	Status 200 OK, usuário XYZ atualizado com sucesso.		Funcional	Alta	Adequação Funcional	Particionamento de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			: false						
US001-CT020	Tentar atualizar usuário com e-mail já utilizado por outro usuário	Usuário XYZ cadastrado, Usuário ABC com email .existe@example.com	ID: XYZ, Email: email .existe@example.com	Status 400 Bad Request, mensagem de erro indicando e-mail já utilizado.		Funcional	Alta	Adequação Funcional (Corretude)	Particionamento de Equivalência
US001-CT021	Criar novo usuário via PUT com ID inexistente	N/A	ID: NOV01D, Nome: Novo Usuário PUT, Email: novoput@example.com, Password: senha PUT, Admini	Status 201 Created, novo usuário NOV01D criado com sucesso.		Funcional	Alta	Adequação Funcional	Particionamento de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			strador : true						
US001-CT022	Tentar criar novo usuário via PUT com e-mail já utilizado	Usuário usuário.existe@example.com já cadastrado	ID: NOV01D2, Nome: Novo Usuário PUT2, Email: usuário.existe@example.com, Password: senha PUT2, Administrador : false	Status 400 Bad Request, mensagem de erro indicando e-mail já utilizado.		Funcional	Alta	Adequação Funcional (Corretude)	Particionamento de Equivalência

8.1.4. Cenário: Exclusão de Usuário

Objetivo: Validar a exclusão bem-sucedida de usuários e a rejeição de exclusão de usuários inexistentes ou com carrinhos associados, garantindo a Adequação Funcional (Corretude) e Integridade (prevenção de exclusão de dados relacionados).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US001-CT023	Excluir usuário existente com sucesso	Usuário com ID XYZ cadastrado e sem carrinho associado	ID: XYZ	Status 200 OK, usuário XYZ excluído com sucesso.		Funcional	Alta	Adequação Funcional	Participante de Equivalência
US001-CT024	Tentar excluir usuário inexistente	N/A	ID: ABC (inexistente)	Status 400 Bad Request (ou 404 Not Found), mensagem de erro indicando usuário não encontrado.		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência
US001-CT025	Tentar excluir usuário com carrinho	Usuário com ID XYZ cadastrado e com carrinho	ID: XYZ	Status 400 Bad Request, mensagem		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	associado	o associado		de erro indicando que o usuário possui carrinho.					

8.2. Cenários e Casos de Teste para US 002 - Login

Esta seção detalha os cenários e casos de teste para a funcionalidade de login, incluindo autenticação bem-sucedida e falhas de autenticação, além da validação do token, garantindo a Adequação Funcional (Corretude) e Segurança (Autenticidade, Confidencialidade, Não Repúdio).

8.2.1. Cenário: Autenticação de Usuário

Objetivo: Validar o processo de autenticação de usuários, incluindo cenários de sucesso e falha, e a geração de token, garantindo a Adequação Funcional e Segurança.

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US002-CT001	Autenticar usuário existente com credenciais válidas	Usuário usuário.válido@example.com com senha	Email: usuário.válido@example.com, Password:	Status 200 OK, token Bearer gerado e retornado.		Funcional	Alta	Adequação Funcional, Segurança (Autenticidade)	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
		senha Valida cadastrado	senha Valida						
US002-CT002	Tentar autenticar usuário não cadastrado	N/A	Email: usuario.nao.cadastrado@example.com, Password: qualquerSenha	Status 401 Unauthorized, mensagem de erro indicando credenciais inválidas.		Funcional	Alta	Adequação Funcional (Correção), Segurança (Autenticidade)	Participação de Equivalência
US002-CT003	Tentar autenticar usuário com senha inválida	Usuário registrado	Email: usuario.valido@example.com, Password: senha Invalida	Status 401 Unauthorized, mensagem de erro indicando credenciais inválidas.		Funcional	Alta	Adequação Funcional (Correção), Segurança (Autenticidade)	Participação de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US002-CT004	Tentar autenticar com e-mail vazio	N/A	Email: "", Password: senha Valida	Status 400 Bad Request, mensagem de erro indicando campo email obrigatório/inálido.		Funcional	Média	Adequação Funcional (Completeness)	Participação de Equivalência
US002-CT005	Tentar autenticar com senha vazia	N/A	Email: usuario.valido@example.com, Password:	Status 400 Bad Request, mensagem de erro indicando campo password obrigatório/inálido.		Funcional	Média	Adequação Funcional (Completeness)	Participação de Equivalência
US002-CT006	Validar duração do token	Usuário autenticado	Token gerado	Token deve ser válido	Necessidade de teste	Funcional	Alta	Segurança (Confidentiality)	Teste de Decisão

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	(10 minutos)	com sucesso		por 10 minutos e inválido após esse período.	de tempo.			dencialidade)	
US002-CT007	Tentar acessar rota protegida com token expirado	Token expirado	Requisição a endpoint protegido com token expirado	Status 401 Unauthorized, mensagem de erro indicando token expirado.		Funcional	Alta	Segurança (Confidencialidade)	Teste de Decisão
US002-CT008	Tentar acessar rota protegida sem token	N/A	Sem token	Status 401 Unauthorized, mensagem de erro indicando token ausente.		Funcional	Alta	Segurança (Confidencialidade)	Participação de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US002-CT009	Tentar acessar rota protegida com token inválido	N/A	Token inválido (ex: modificado)	Status 401 Unauthorized, mensagem de erro indicando token inválido.		Funcional	Alta	Segurança (Confidencialidade)	Participação de Equivalência

8.3. Cenários e Casos de Teste para US 003 - Produtos

Esta seção detalha os cenários e casos de teste para a funcionalidade de gerenciamento de produtos, incorporando as características de qualidade da ISO/IEC 25010:2011 e as técnicas de design de teste.

8.3.1. Cenário: Cadastro de Novo Produto

Objetivo: Validar o cadastro bem-sucedido de novos produtos com dados válidos e a rejeição de cadastros com dados inválidos ou duplicados, garantindo a Adequação Funcional (Completeness, Correctness, Appropriability) e Segurança (Integrity).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US003-CT001	Cadastrar produto com todos	Usuário autenticado	Nome: Produto Teste 1,	Status 201 Created, produto		Funcional	Alta	Adequação Funcional	Participação de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	os campos válidos (usuário autenticado)	(token válido)	Preço: 100, Descrição: Descrição do Produto 1, Quantidade: 10	o cadastrado com sucesso.					
US003-CT002	Tentar cadastrar produto com nome já utilizado	Usuário autenticado, Produto Produto Existente já cadastrado	Nome: Produto Existente, Preço: 50, Descrição: Nova Descrição, Quantidade: 5	Status 400 Bad Request, mensagem de erro indicando nome de produto já utilizado.		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência
US003-CT003	Tentar cadastrar produto sem autenticação	N/A	Nome: Produto Sem Auth, Preço: 20,	Status 401 Unauthorized, mensagem		Funcional	Alta	Adequação Funcional (Corretude), Segur	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			Descrição: Descrição, Quantidade: 2	de erro indicando token ausente/inválido.				anção (Confidencialidade)	
US003-CT004	Tentar cadastrar produto com campo nome vazio	Usuário autenticado	Nome: ``, Preço: 10, Descrição: `Descrição`, Quantidade: `1`	Status 400 Bad Request, mensagem de erro indicando campo nome obrigatório.		Funcional	Média	Adequação Funcional (Completeness)	Participação de Equivalência
US003-CT005	Tentar cadastrar produto com campo preço inválido (não numérico)	Usuário autenticado	Nome: Produto Inválido, Preço: abc, Descrição: Descrição, Quantidade: 1	Status 400 Bad Request, mensagem de erro indicando tipo de dado inválido para		Funcional	Média	Adequação Funcional (Correctness)	Participação de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
				preco.					
US003-CT006	Tentar cadastrar produto com campo quantidade inválido (não numérico)	Usuário autenticado	Nome: Produto Inválido 2, Preço: 10, Descrição: Descrição, Quantidade: xyz	Status 400 Bad Request, mensagem de erro indicando o tipo de dado inválido para quantidade.		Funcional	Média	Adequação Funcional (Corretude)	Participante de Equivalência

8.3.2. Cenário: Listagem e Busca de Produtos

Objetivo: Validar a correta listagem e busca de produtos cadastrados, garantindo a Adequação Funcional (Completeness, Correctness) e Eficiência de Desempenho (Performance).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US003-CT007	Listar todos os produtos	Vários produtos cadastrados	N/A	Status 200 OK, lista de produtos		Funcional	Alta	Adequação Funcional	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	cadastros			os conteúdos de todos os produtos cadastrados.					
US003-CT008	Listar produtos quando não há nenhum produto cadastrado	Nenhum produto cadastrado	N/A	Status 200 OK, lista de produtos vazia.		Funcional	Média	Adequação Funcional	Participante de Equivalência
US003-CT009	Buscar produto por ID válido	Produto com ID XYZ cadastrado	ID: XYZ	Status 200 OK, dados do produto XYZ retornados.		Funcional	Alta	Adequação Funcional	Participante de Equivalência
US003-CT010	Buscar produto por ID	N/A	ID: ABC (inexistente)	Status 400 Bad Request		Funcional	Alta	Adequação Funcional	Participante de

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	inexistente			st (ou 404 Not Found), mensagem de erro indicando produto não encontrado.				(Corretude)	Equivalência

8.3.3. Cenário: Atualização de Produto

Objetivo: Validar a atualização de dados de produtos existentes e a criação de novos produtos via PUT quando o ID não existe, garantindo a Adequação Funcional (Corretude, Completude) e Segurança (Integridade).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US003-CT011	Atualizar dados de produto existente com sucesso	Usuário autenticado, Produto com ID XYZ cadastrado	ID: XYZ, Nome: Produto Atualizado, Preço: 150,	Status 200 OK, produto XYZ atualizado com sucesso.		Funcional	Alta	Adequação Funcional	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			Descrição: Nova Descrição, Quantidade: 15						
US003-CT012	Tentar atualizar produto com nome já utilizado por outro produto	Usuário autenticado, Produto XYZ cadastrado, Produto ABC com Nome Existente	ID: XYZ, Nome: Nome Existente	Status 400 Bad Request, mensagem de erro indicando nome de produto já utilizado.		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência
US003-CT013	Criar novo produto via PUT com ID inexistente	Usuário autenticado	ID: NOV01D, Nome: Novo Produto PUT, Preço: 200, Descrição:	Status 201 Created, novo produto NOV01D criado com		Funcional	Alta	Adequação Funcional	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			ção: Descrição PUT, Quantidade: 20	sucesso.					
US003-CT014	Tentar criar novo produto via PUT com nome já utilizado	Usuário autenticado, Produto Produto Existente já cadastrado	ID: NOV01D2, Nome: Produto Existente, Preço: 250, Descrição: Descrição PUT 2, Quantidade: 25	Status 400 Bad Request, mensagem de erro indicando nome de produto já utilizado.		Funcional	Alta	Adequação Funcional (Corretude)	Participante de Equivalência

8.3.4. Cenário: Exclusão de Produto

Objetivo: Validar a exclusão bem-sucedida de produtos e a rejeição de exclusão de produtos inexistentes ou com carrinhos associados, garantindo a Adequação Funcional (Corretude) e Integridade (prevenção de exclusão de dados relacionados).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
US003-CT015	Excluir produto existente com sucesso	Usuário autenticado, Produto com ID XYZ cadastrado e sem carrinho associado	ID: XYZ	Status 200 OK, produto XYZ excluído com sucesso.		Funcional	Alta	Adequação Funcional	Participante de Equivalência
US003-CT016	Tentar excluir produto inexistente	Usuário autenticado	ID: ABC (inexistente)	Status 400 Bad Request (ou 404 Not Found), mensagem de erro indicando produto não encontrado.		Funcional	Alta	Adequação Funcional (Correção)	Participante de Equivalência
US003-CT017	Tentar excluir produto	Usuário autenticado	ID: XYZ	Status 400 Bad		Funcional	Alta	Adequação Funcional	Participante de

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
	o com carrinho associado	cado, Produto com ID XYZ cadastrado e com carrinho associado		Requerst, mensagem de erro indicando que o produto está em um carrinho.				nal (Corretude)	Equivalência

8.5. Cenários e Casos de Teste Não-Funcionais

Esta seção detalha os cenários e casos de teste para os aspectos não-funcionais da API, alinhados com as características de qualidade da ISO/IEC 25010:2011.

8.5.1. Cenário: Testes de Eficiência de Desempenho

Objetivo: Avaliar o comportamento de tempo, utilização de recursos e capacidade da API sob diferentes cargas, garantindo a Eficiência de Desempenho.

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
NFR-PERF-C T001	Teste de carga para endpoint de login	100 usuários simultâneos	Credenciais válidas para 100 usuários	Tempo de resposta médio < 500ms	Utilizar k6 para simular carga.	Não-Funcional (Performance)	Alta	Eficiência de Desempenho (Comportamento de	Análise de Valor Limite

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
				para 95% das requisições.				Tempo, Capacidade)	
NFR-PERF-C T002	Teste de carga para listagem de produtos	500 usuários simultâneos	N/A	Tempo de resposta médio < 500ms para 95% das requisições.	Utilizar k6 para simular carga.	Não-Funcional (Performance)	Alta	Eficiência de Desempenho (Comportamento de Tempo, Capacidade)	Análise de Valor Limite
NFR-PERF-C T003	Teste de estresse para cadastro de usuários	Aumento gradual da carga até falha	Dados de cadastro de usuários válidos	Identificar o ponto de ruptura da API.	Utilizar k6 para simular estresse.	Não-Funcional (Performance)	Média	Eficiência de Desempenho (Capacidade)	Análise de Valor Limite

8.5.2. Cenário: Testes de Segurança

Objetivo: Identificar vulnerabilidades de segurança na API, incluindo injeção, autenticação e autorização, garantindo a Segurança (Confidencialidade, Integridade, Não Repúdio, Responsabilidade, Autenticidade).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
NFR-SEC-CT 001	Teste de injeção de SQL em campos de entrada (ex: email, nome)	N/A	\\ ' OR \\ ' 1 \\ ' = \\ ' 1 em campos de texto	API deve rejeitar a requisição ou sanitizar a entrada, sem erro de banco de dados.	Utilizar OWASP ZAP ou ferramentas de injeção.	Não-Funcional (Segurança)	Alta	Segurança (Integridade)	Participante de Equivalência
NFR-SEC-CT 002	Teste de força bruta no endpoint de login	N/A	Múltiplas tentativas de login com senhas inválidas	API deve bloquear o acesso após N tentativas ou introduzir delay.		Não-Funcional (Segurança)	Alta	Segurança (Autenticidade)	Análise de Valor Limite
NFR-SEC-CT 003	Teste de acesso a recursos protegidos sem token	N/A	Requisição a endpoint protegido sem token de	Status 401 Unauthorized.		Não-Funcional (Segurança)	Alta	Segurança (Confidencialidade)	Participante de Equivalência

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
			autenticação						
NFR-SEC-CT004	Teste de acesso a recursos protegidos com token expirado	Token expirado	Requisição a endpoint protegido com token expirado	Status 401 Unauthorized.		Funcional, Não-Funcional (Segurança)	Alta	Segurança (Confidencialidade)	Teste de Decisão
NFR-SEC-CT005	Teste de acesso a recursos de administrador por usuário comum	Usuário comum autenticado	Requisição a endpoint de administrador com token de usuário comum	Status 403 Forbidden.		Funcional, Não-Funcional (Segurança)	Alta	Segurança (Responsabilidade)	Teste de Decisão

8.5.3. Cenário: Testes de Compatibilidade

Objetivo: Garantir que a API funcione corretamente em diferentes ambientes e versões de software, garantindo a Compatibilidade (Coexistência, Interoperabilidade).

ID do Caso de Teste	Descrição do Caso de Teste	Pré-condições	Dados de Entrada	Resultado Esperado	Observações	Tipo de Teste	Prioridade	Característica de Qualidade	Técnica de Design
NFR-COMP-CT001	Teste de compatibilidade com diferentes versões do Node.js	API implantada em Node.js v14 e v16	Requisições a todos os endpoints	API deve responder corretamente em ambas as versões.	Requer ambiente de teste com múltiplas versões de Node.js.	Não-Funcional (Compatibilidade)	Média	Compatibilidade (Coexistência)	Particionamento de Equivalência
NFR-COMP-CT002	Teste de compatibilidade com diferentes tipos de clientes HTTP	N/A	Requisições via Postman, cURL, bibliotecas Python/Java	API deve responder corretamente a todos os clientes.		Não-Funcional (Compatibilidade)	Baixa	Compatibilidade (Interoperabilidade)	Particionamento de Equivalência

9. Priorização da Execução dos Cenários de Teste

A priorização da execução dos cenários de teste será baseada em uma combinação de fatores, incluindo a criticidade da funcionalidade, o risco associado, a frequência de uso esperada e a dependência de outros módulos. A prioridade de cada caso de teste já está indicada nas tabelas de cenários (Alta, Média, Baixa).

- Prioridade Alta: Casos de teste que cobrem funcionalidades críticas, cenários de alto risco (segurança, falhas que impedem o uso principal do sistema) e fluxos de usuário principais. Estes serão executados primeiro e com maior frequência.

- Prioridade Média: Casos de teste que cobrem funcionalidades importantes, cenários de risco moderado e fluxos alternativos. Serão executados após os de alta prioridade.
- Prioridade Baixa: Casos de teste que cobrem funcionalidades menos críticas, cenários de baixo risco ou casos de borda menos prováveis. Serão executados quando houver tempo e recursos disponíveis.

10. Matriz de Risco

Risco	Impacto	Probabilidade	Mitigação	Contingência
Indisponibilidade do Servidor da API	Testes bloqueados, atraso no cronograma	Média	Monitoramento contínuo do ambiente, comunicação proativa com a equipe de desenvolvimento /infraestrutura.	Utilizar ambiente de mock local para testes críticos, se possível.
Dados Inconsistentes no Ambiente de Teste	Resultados de teste não confiáveis, falsos positivos/negativos	Média	Implementação de rotinas de reset de dados, uso de massa de dados controlada e isolada por teste.	Recriação do ambiente de teste, execução de testes de fumaça após reset.
Mudanças Frequentes na API sem Aviso Prévio	Casos de teste desatualizados, retrabalho	Baixa	Comunicação constante com a equipe de desenvolvimento , uso de versionamento de API.	Atualização rápida dos casos de teste, priorização de testes de regressão.
Limitações de Massa de Teste	Cobertura de teste incompleta, dificuldade em replicar cenários	Média	Geração de massa de dados automatizada, uso de ferramentas de	Identificação de cenários críticos que podem ser impactados, aceitação de

Risco	Impacto	Probabilidade	Mitigação	Contingência
			virtualização de dados.	cobertura limitada em cenários de baixa prioridade.

11. Referências

Este plano de testes foi elaborado com base nas seguintes referências:

- ISTQB Certified Tester Foundation Level (CTFL) 4.0 Syllabus: Guia para princípios de teste, processo de teste e técnicas de design de teste.
- ISO/IEC 25010:2011 – System and Software Quality Models: Define características de qualidade de software para avaliação e categorização de requisitos não funcionais.
- ISO/IEC/IEEE 29119 – Software Testing: Série de padrões internacionais para processos e documentação de teste de software.
- IEEE 829 – Test Documentation: Padrão para a estrutura e conteúdo da documentação de teste.