### Relatório de Testes Baseado em Sessão (SBTM) - API ServeRest

Nome da Validação das APIs de Usuários, Login e Produtos (US 001,

Sessão/Charter: 002, 003)

ID da Sessão:

**Testador(a):** Victor Lohan Sousa De Assis

**Data:** 15/08/2025

**Duração Planejada:** 180 minutos

**Duração Real:** 240 minutos

### 1. Apresentação

Este documento detalha o planejamento, execução e análise da sessão de testes realizada sobre a API ServeRest. O foco da sessão foi a validação funcional e de regras de negócio das User Stories US 001 - Usuários, US 002 - Login e US 003 - Produtos. Todos os testes foram estruturados e automatizados utilizando a ferramenta Postman.

## 2. Objetivo

O objetivo principal desta sessão de testes é garantir que as funcionalidades de CRUD de Usuários, Autenticação e CRUD de Produtos estejam em conformidade com os Critérios de Aceitação definidos. Busca-se validar tanto os fluxos de sucesso quanto os cenários de exceção e erro, assegurando a robustez e a confiabilidade da API.

### 3. Escopo

Funcionalidades DENTRO do Escopo:

- API de Usuários (/usuarios): Criação, Listagem (geral e por ID), Atualização e Exclusão.
- API de Login (/login): Autenticação de usuários e geração de token Bearer.
- API de Produtos (/produtos): Criação, Listagem, Atualização e Exclusão, utilizando o token de autenticação.

#### 4. Análise

A estratégia de teste foi definida a partir da análise detalhada das User Stories e seus Critérios de Aceitação. Cada critério foi decomposto em um ou mais cenários de teste específicos. A análise seguiu os seguintes passos:

- 1. Mapeamento de Requisitos: Cada Critério de Aceitação foi mapeado para um endpoint específico da API (ex: "Não deve ser possível criar um usuário com e-mail já utilizado" mapeado para POST /usuarios).
- 2. Identificação de Fluxos: Foi desenhado um fluxo de ponta a ponta para simular a jornada de um usuário real: Cadastro -> Login -> Gestão de Produtos.
- 3. Definição de Dados de Teste: Foram utilizados dados dinâmicos (variáveis do Postman como {{\$randomEmail}}) para garantir a independência dos testes automatizados.

### 5. Técnicas Aplicadas

- Teste de API: Validação direta dos endpoints, verificando status codes, headers e o corpo das respostas (payloads).
- Teste Funcional: Foco em garantir que cada funcionalidade (CRUD) opera conforme o esperado.
- Partição de Equivalência: Agrupamento de entradas para reduzir o número de casos de teste.
- Análise de Valor Limite: Aplicada em regras como o comprimento da string...
- Teste de Estado/Fluxo: Validação da sequência de estados da aplicação (ex: um usuário não pode gerenciar produtos antes de estar autenticado).

# 6. Cenários de Teste Planejados

ID do		
Cenário	User Story	Descrição do Cenário
CT-001	US 001	Cadastrar um novo usuário com dados válidos e e-mail permitido.
CT-002	US 001	Tentar cadastrar um usuário com um e-mail já existente.
CT-003	US 001	Tentar cadastrar um usuário com e-mail de provedor não permitido (gmail/hotmail).
CT-004	US 001	Listar todos os usuários cadastrados.
CT-005	US 001	Atualizar os dados de um usuário existente.
CT-006	US 001	Tentar atualizar um usuário usando um ID inexistente (deve criar um novo).
CT-007	US 001	Deletar um usuário existente com sucesso.
CT-008	US 002	Realizar login com credenciais válidas.
CT-009	US 002	Tentar realizar login com senha incorreta.
CT-010	US 002	Tentar realizar login com um e-mail não cadastrado.
CT-011	US 003	Cadastrar um novo produto com sucesso (autenticado).
CT-012	US 003	Tentar cadastrar um produto sem token de autenticação.
CT-013	US 003	Tentar cadastrar um produto com nome já existente.
CT-014	US 003	Excluir um produto com sucesso (autenticado).

# 7. Priorização da Execução dos Cenários de Teste

Prioridade	Critério	Cenários
Alta	Fluxos críticos e caminho feliz que habilitam outras funcionalidades	CT-001, CT-008, CT-011
Média	Cenários de erro comuns e validações de regras de negócio importantes.	CT-002, CT-005, CT-007, CT-009, CT-010, CT-013, CT-014
Baixa	Casos de borda e cenários de erro menos frequentes.	CT-003, CT-004, CT-006, CT-012

## 8. Matriz de Risco

ID do Risco	Descriçã o do Risco	Probabili dade	Impacto	Nível do Risco	Ação de Mitigação
RISCO- 01	Acesso não autoriza do a rotas protegid as (ex: /produto s).	Baixa	Alto	Médio	Teste CT-012 que tenta acessar a rota sem token, esperando um erro 401.
RISCO- 02	Criação de dados inconsist entes (ex: usuários com mesmo e-mail).	Baixa	Médio	Baixo	Teste CT-002 que valida a trava de e-mail duplicado.
RISCO- 03	Falha na autentic ação não retorna erro claro para o usuário.	Média	Médio	Médio	Testes CT-009 e CT-010 que validam o status code 401 e a mensagem de erro.
RISCO- 04	Regras de negócio opcionai s (ex: bloqueio de provedor de	Alta	Baixo	Médio	Teste CT-003 foi criado para validar essa regra. Mesmo que falhe, evidencia a lacuna.

e-mail)
não são
impleme
ntadas.

### 9. Cobertura de Testes

A cobertura de testes foi medida com base nos Critérios de Aceitação (CA) de cada User Story. Segue tabela de relacionamento:

User Story	Critério de Aceitação	Cenário de Teste Correspondent e	Coberto?
US 001	Campos NOME, E-MAIL, PASSWORD e ADMINISTRAD OR	CT-001	Sim
US 001	Não permitir e-mail já utilizado (POST)	CT-002	Sim
US 001	Não permitir e-mail já utilizado (PUT)	[Similar ao CT-002, mas com PUT]	Sim
US 001	Criar novo usuário se ID não existe no PUT	CT-006	Sim
US 001	Bloquear e-mails de provedor gmail/hotmail	CT-003	Sim
US 002	Usuários não cadastrados não autenticam	CT-010	Sim
US 002	Senha inválida não autentica	CT-009	Sim
US 002	Retornar status 401 em falhas	CT-009, CT-010	Sim
US 002	Autenticação gera token Bearer	CT-008	Sim

US 003	Usuários não autenticados não realizam ações	CT-012	Sim
US 003	Não cadastrar produtos com nomes duplicados	CT-013	Sim

## 10. Resultados da Execução e Observações

A suíte de testes automatizados foi executada através do Postman Collection Runner. A execução seguiu a ordem de priorização, validando os fluxos principais antes dos casos de borda.

ID do Cenário	Descrição do Cenário	Status	Observações / ID do Defeito
CT-001	Cadastrar um novo usuário com dados válidos.	Passou	Usuário criado com sucesso. ID e credenciais salvas para os próximos testes.
CT-002	Tentar cadastrar um usuário com um e-mail já existente.	Passou	A API retornou corretamente o erro 400 com a mensagem esperada.
CT-004	Listar todos os usuários cadastrados.	Passou	A lista de usuários foi retornada com sucesso.
CT-005	Atualizar os dados de um usuário existente.	Passou	Os dados do usuário foram atualizados com sucesso.
CT-006	Tentar atualizar usuário com ID inexistente (deve criar novo).	Passou	A API corretamente criou um novo registro e retornou o status 201.
CT-007	Deletar um usuário existente com sucesso.	Passou	Usuário de teste foi removido da base de dados ao final do fluxo.
CT-008	Realizar login com credenciais válidas.	Passou	Login bem-sucedido. Token de autenticação gerado e salvo na variável authToken.

CT-009	Tentar realizar login com senha incorreta.	Passou	A API retornou corretamente o erro 401 (Unauthorized).
CT-010	Tentar realizar login com um e-mail não cadastrado.	Passou	A API retornou corretamente o erro 401 com a mensagem esperada.
CT-011	Cadastrar um novo produto com sucesso (autenticado).	Passou	Produto criado com sucesso utilizando o token de autenticação.
CT-012	Tentar cadastrar um produto sem token de autenticação.	Passou	Acesso negado com status 401 e mensagem de token ausente, conforme esperado.
CT-013	Tentar cadastrar um produto com nome já existente.	Passou	A API retornou corretamente o erro 400 com a mensagem de nome duplicado.
CT-014	Excluir um produto com sucesso (autenticado).	Passou	Produto de teste foi removido com sucesso.

## 11. Testes Candidatos a Automação

Dada a natureza da API, 100% dos cenários de teste planejados foram identificados como candidatos ideais e implementados como testes automatizados no Postman.

#### Justificativa:

- **Regressão:** Garante que novas implementações não quebrem as funcionalidades principais de usuário, login e produto.
- **Velocidade:** A execução completa da suíte de testes leva segundos, fornecendo feedback rápido para a equipe de desenvolvimento.
- **Confiabilidade:** A automação elimina o risco de erro humano na execução de testes repetitivos.