**Interface gráfica do usuário, Texto, Site

O conteúdo gerado por IA pode estar incorreto.**

**Documentação de Teste**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Data e Hora do Início** | **QA Responsável** | |  | | --- | | **Funcionalidade** |  |  | | --- | |  | | | **Referência** | | --- |  |  | | --- | |  | |
| 10 /11/2025 15:48 PM | Jefferson Paulo | Login de Empreendedores | Plano\_e\_Estrategia\_de\_Testes\_Adaptada\_SGVC.docx / TestCharter\_Login\_Empreendedores.docx |

**SGVC – Documento Completo de Cenários de Teste BDD (Integra API + WEB + A11Y + Responsividade + Segurança + K6 + Manuais)**

**User Story 001 – Login de Empreendedores-**

Link: [User Story-001–SGVC- Login de Empreendedores](https://github.com/jeff-barbosa123/Software-para-Gest-o-de-Vendas-e-Clientes/issues/2)

**Autor:** Jefferson Paulo – QA  
**Versão:** 2.0

|  |  |  |  |
| --- | --- | --- | --- |
| MAPA DE REQUISITOS – User Story 001 (Login de Empreendedores) | | | |
| ID | **Tipo** | **Requisito** | **Descrição** |
| REQ-001 | Funcional | |  | | --- | |  |  |  | | --- | | Autenticação | | |  | | --- | |  |   O usuário deve conseguir autenticar-se com credenciais válidos |
| REQ-002 | Funcional | Validação | A API deve validar campos, formatos, espaços, trim, UTF-8 e dados longos. |
| REQ-003 | Funcional | Segurança | A API deve prevenir ataques, SQL Injection, tentativas múltiplas, token inválido, brute-force |
| REQ-004 | Funcional | Sessão/Token | A API deve gerenciar expiração de sessão, tempo limite e validação de token |
| REQ-005 | Funcional | Feedback/Erros | A API e futura interface devem exibir menagem claras e padronizadas |
| REQ-006 | Funcional | Robustez/Falhas | A API deve lidar com indisponibilidade, timeouts e perdas de conexão. |
| REQ-007 | Não-funcional | Acessibilidade | A tela deve seguir WCAG 2.1 (contraste, ARIA, teclado, foco, zoom 200%). |
| REQ-008 | Não-funcional | Responsividade | A UI deve funcionar em mobile, tablet, desktop e landscape sem quebra. |
| REQ-009 | Não-funcional | |  | | --- | | Segurança Avançada |  |  | | --- | |  | | Deve impedir XSS, manipulação de headers e payloads maliciosos. |
| REQ-010 | Não-funcional | Usabilidade | A UI deve carregar em < 2s, mostrar loader e feedback imediato. |
| REQ-011 | Não-funcional | Confiabilidade | O sistema deve manter consistência após reconexão, falhas internas e retries. |

|  |  |  |  |
| --- | --- | --- | --- |
| TABELA DE RASTREABILIDADE COMPLETA | | | |
| Requisito | **Condição** | **Cenário BDD** | **Heuristica** **/ Técnica** |
| REQ-OO1 | Credenciais válidas | Login com credenciais válidas | VADER – Data |
| REQ-001 | Credenciais inválidas | Senha incorreta | VADER – Errors |
| REQ-001 | E-mail inexistente | Rejeitar e-mail inexistente | VADER – Data |
| REQ-001 | Fluxo feliz | Autenticar com sucesso | Fluxo Feliz (Técnica) |
| REQ-002 | E-mail vazio | Campo e-mail obrigatório | Campos Vazios |
| REQ-002 | Senha Vazia | Campo senha obrigatório | Campos Vazios |
| REQ-002 | E-mail Sem @ | Validar e-mail sem arroba | Validação |
| REQ-002 | Senha Curta | Senha com menos de 3 caracteres | Valor Limite |
| REQ-002 | Espaços | Normalização (trim) | Espaços Extras |
| REQ-002 | Strings Longas | E-mail com 255 + caracteres | String Longas |
| REQ-002 | Unicode | UTF-8 (acentos, chinês, árabe) | Codificação/Unicode |
| REQ-003 | Ataques SQL | SQL Injection | SQL Injection (Heurísticas de Ataque) |
| REQ-003 | Ataques SQL 2 | SQL na senha | SQL Injection |
| REQ-003 | Falhas Agrupadas | Tentativas alteradas | Falhas Agrupadas |
|  |  |  |  |
| REQ-003 | Bloqueio | Bloqueio após 3 tentativas | Segurança (Categoria) |
| REQ-004 | Sessão Expira | Token expirado | Tempo Limite |
| REQ-004 | Inatividade | Expiração após 30 minutos | VADER – Responsiveness |
| REQ-005 | Mensagens Claras | Erro visual | VADER – Errors |
| REQ-006 | API Down | API 503 | Interrupções |
| REQ-006 | Perda de conexão | Conexão Perdida | |  | | --- | | VADER – Responsiveness | |
| REQ-011 | Falha durante venda | Interrupção no processo | Robustez |
| REQ-011 | Reinício servidor | Recuperação após restart | Confiabilidade |
| REQ-007 | Acessibilidade Básica | Navegação por teclado | Acessibilidade |
| REQ-007 | Leitor de tela | ARIA Roles | Acessibilidade WCAG |
| REQ-007 | Contraste | WCAG AA 4.5:1 | Acessibilidade |
| REQ-007 | Semântica | HTML semântico | Acessibilidade |
| REQ-007 | Foco visível | Indicador de foco | WCAG |
| REQ-007 | Zoom 200% | Zoom permitido | WCAG |
| REQ-007 | Axe DevTools | Nenhum erro critico | Acessibilidade (Ferramenta) |
| REQ-008 | Responsividade  Mobile | 360px | Layout Responsivo |
| REQ-008 | Responsividade Tablet | 768px | Layout Responsivo |
| REQ-008 | Responsividade Desktop | 1920px | Layout Responsivo |
| REQ-008 | Quebra de layout | Textos e campos visíveis | Layout |
| REQ-008 | Landscape | Rotação | Adaptabilidade |
| REQ-010 | Usabilidade | Painel <2s | Performance UX |
| REQ-010 | Legibilidade | Gráficos claros | UX |
| REQ-010 | Botão visível | CTA sempre acessível | UX |
| REQ-010 | Loader | Feedback claro | UX |

**CENÁRIOS AUTOMATIZÁVEIS + MANUAIS**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABELA CONSOLIDADA DE TODOS OS CENÁRIOS- API — SUPERTES + MOCHA | | | | |
| ID | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| API-001 | API | Login com credenciais válidas | Automático | SuperTest + Mocha |
| API-002 | API | Senha Incorreta   |  | | --- | |  | | Automático | SuperTest + Mocha |
| API-003 | API | E-mail inexistente | Automático | SuperTest + Mocha |
| API-004 | API | E-mail sem @ | Automático | SuperTest + Mocha |
| API-005 | API | Senha com 2 caracteres | Automático | SuperTest + Mocha |
| API-006 | API | Campo e-mail vazio | Automático | SuperTest + Mocha |
| API-007 | API | Campo senha vazio | Automático | SuperTest + Mocha |
| API-008 | API | Trim em e-mail | Automático | SuperTest + Mocha |
| API-009 | API | Trim em senha | Automático | SuperTest + Mocha |
| API-010 | API | E-mail 255+ caracteres | Automático | SuperTest + Mocha |
| API-011 | API | Senha 255+ caracteres | Automático | SuperTest + Mocha |
| API-012 | API | UTF-8 acentos | Automático | SuperTest + Mocha |
| API-013 | API | UTF-8 chinês | Automático | SuperTest + Mocha |
| API-014 | API | UTF-8 árabe | Automático | SuperTest + Mocha |
| API-015 | API | SQL Injection no e-mail | Automático | Postman / SuperTest |
| API-016 | Segurança | SQL Injection na senha | Automático | Postman |
| API-017 | Segurança | Tentativas alternadas | Semi-automático | SuperTest |
| API-018 | API | Bloqueio 3 tentativas | Automático | SuperTest |
| API-019 | API | Login durante bloqueio | Automático | SuperTest |
| API-020 | API | Login após desbloqueio | Automático | SuperTest |
| API-021 | API | Token expirado | Automático | SuperTest |
| API-022 | API | Sessão expira por inatividade | Semi-automático | SuperTest |
| API-023 | API | Acesso sem token | Automático | SuperTest |
| API-024 | API | API indisponível (503) | Manual | Exploratório |
| API-025 | Robustez | Perda de conexão | Manual | Exploratório |
| API-026 | Robustez | Erro interno (500) | Automático | SuperTest |
| API-027 | API | JSON Inválido | Automático | SuperTest |

**CASOS DE TESTE BDD – API (SuperTest + Mocha)**

**API-001 – Login com credenciais válidas**

**Cenário: Autenticar usuário com credenciais válidas**

Dado que existe um usuário cadastrado com e-mail e senha válidos  
Quando envio uma requisição POST para /auth/login com as credenciais corretas  
Então a API deve retornar o status 200  
E deve retornar um objeto contendo um token JWT válido  
E o token deve conter os campos sub, iat, exp

**API-002 – Senha incorreta**

**Cenário: Rejeitar login com senha incorreta**

Dado que existe um usuário cadastrado  
Quando envio POST /auth/login com e-mail válido e senha incorreta  
Então a API deve retornar status 401  
E deve retornar a mensagem "Credenciais inválidas"

**API-003 – E-mail inexistente**

**Cenário: Rejeitar login com e-mail não cadastrado**

Dado que o usuário não existe na base  
Quando envio POST /auth/login com um e-mail não cadastrado  
Então a API deve retornar status 401  
E deve retornar a mensagem "Credenciais inválidas"

**API-004 – E-mail sem @**

**Cenário: Rejeitar formato inválido de e-mail**

Dado que envio um e-mail no formato incorreto "usuario.com"  
Quando envio POST /auth/login com esse e-mail inválido  
Então a API deve retornar status 400  
E deve retornar a mensagem "Formato de e-mail inválido"

**API-005 – Senha com 2 caracteres**

**Cenário: Rejeitar senha muito curta**

Dado que forneço uma senha com 2 caracteres  
Quando envio POST /auth/login  
Então a API deve retornar status 400  
E deve retornar a mensagem "Senha muito curta"

**API-006 – Campo e-mail vazio**

**Cenário: Rejeitar login sem e-mail**

Dado que não informo um e-mail  
Quando envio POST /auth/login  
Então a API deve retornar status 400  
E deve retornar "E-mail é obrigatório"

**API-007 – Campo senha vazio**

**Cenário: Rejeitar login sem senha**

Dado que não informo a senha  
Quando envio POST /auth/login  
Então a API deve retornar status 400  
E deve retornar "Senha é obrigatória"

**API-008 – Trim em e-mail**

**Cenário: Remover espaços extras do e-mail**

Dado que envio um e-mail com espaços antes e depois  
Quando envio POST /auth/login com " usuario@teste.com "  
Então a API deve aplicar trim()  
E autenticar normalmente se o e-mail for válido

**API-009 – Trim em senha**

**Cenário: Remover espaços extras da senha**

Dado que envio uma senha válida com espaços adicionais  
Quando envio POST /auth/login com " admin123 "  
Então a API deve aplicar trim()  
E autenticar se a senha for correta

**API-010 – E-mail com 255+ caracteres**

**Cenário: Validar e-mail extremamente longo**

Dado que envio um e-mail com mais de 255 caracteres  
Quando envio POST /auth/login  
Então a API não deve quebrar  
E deve retornar erro 400 com mensagem "E-mail muito longo"

**API-011 – Senha com 255+ caracteres**

**Cenário: Validar senha extremamente longa**

Dado que envio uma senha com mais de 255 caracteres  
Quando envio POST /auth/login  
Então a API deve responder com status 400  
E retornar "Senha muito longa"

**API-012 – UTF-8 com acentos**

**Cenário: Validar caracteres acentuados**  
Dado que envio "joão@teste.com" no campo e-mail  
Quando envio POST /auth/login  
Então a API deve manter a integridade da string  
E processar normalmente

**API-013 – UTF-8 com caracteres chineses**

**Cenário: Validar caracteres asiáticos**

Dado que envio um e-mail ou senha com caracteres chineses  
Quando envio POST /auth/login  
Então a API deve aceitar o payload UTF-8  
E não deve corromper caracteres

**API-014 – UTF-8 com caracteres árabes**

**Cenário: Validar caracteres árabes**

Dado que envio senha "مرحبا123"  
Quando envio POST /auth/login  
Então a API deve aceitar o texto UTF-8  
E retornar resposta consistente

**API-015 – SQL Injection no e-mail**

**Cenário: Impedir SQL Injection no campo e-mail**

Dado que envio " ' OR '1'='1 " como e-mail  
Quando envio POST /auth/login  
Então a API deve retornar 401  
E não deve autenticar  
E não deve vazar stacktrace

**API-016 – SQL Injection na senha**

**Cenário: Impedir SQL Injection no campo senha**

Dado que envio a senha "admin'--"  
Quando envio POST /auth/login  
Então a API deve retornar 401  
E registrar a tentativa suspeita

**API-017 – Tentativas alternadas**

**Cenário: Alternar erros de e-mail e senha**

Dado que envio tentativas com e-mail válido e senha errada  
E depois e-mail errado e senha correta  
Quando realizo 3 tentativas inválidas  
Então a API deve manter comportamento consistente  
E não deve autenticar

**API-018 – Bloqueio após 3 tentativas**

**Cenário: Bloquear após 3 falhas consecutivas**

Dado que o usuário erra a senha por 3 vezes  
Quando envia uma quarta tentativa  
Então a API deve retornar 403  
E deve retornar "Usuário temporariamente bloqueado"

**API-019 – Login durante bloqueio**

**Cenário: Impedir login enquanto bloqueado**

Dado que o usuário está bloqueado  
Quando tenta realizar login  
Então a API deve retornar 403  
E exibir "Usuário temporariamente bloqueado"

**API-020 – Login após desbloqueio**

**Cenário: Liberar login depois do tempo de bloqueio**

Dado que o tempo de bloqueio expirou  
Quando envio as credenciais corretas  
Então o login deve ser permitido  
E retornar token JWT válido

**API-021 – Token expirado**

**Cenário: Impedir acesso com token vencido**

Dado que possuo um token JWT expirado  
Quando aceso uma rota protegida /clientes  
Então a API deve retornar status 401  
E mensagem "Token expirado"

**API-022 – Sessão expira por inatividade**

**Cenário: Sessão expira após 30 minutos**  
Dado que estou autenticado  
E fico 30 minutos sem uso  
Quando tento acessar /clientes  
Então a API deve retornar 401  
E exigir novo login

**API-023 – Acesso sem token**

**Cenário: Impedir acesso sem autenticação**

Dado que não envio o header Authorization  
Quando acesso uma rota protegida  
Então a API deve retornar 401  
E exibir "Token não fornecido"

**API-024 – API indisponível (503)**

**Cenário: Servidor fora do ar**

Dado que a API está indisponível  
Quando envio POST /auth/login  
Então devo receber erro 503  
E mensagem "Serviço temporariamente indisponível"

**API-025 – Perda de conexão**

**Cenário: Falha de conexão com a internet**

Dado que minha rede está instável  
Quando envio uma requisição  
Então o cliente deve exibir erro de rede  
E não deve quebrar a aplicação

**API-026 – Erro interno (500)**

**Cenário: Tratar erro inesperado no servidor**

Dado que ocorre uma exceção interna  
Quando envio POST /auth/login  
Então a API deve retornar status 500  
E uma mensagem genérica "Erro interno"  
E não deve expor stacktrace

**API-027 – JSON Inválido**

**Cenário: Payload mal formatado**

Dado que envio um JSON inválido no body  
Quando envio POST /auth/login  
Então a API deve retornar status 400  
E mensagem "JSON mal formatado"

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABELA CONSOLIDADA DE TODOS OS CENÁRIOS- WEB – CYPRESS UI | | | | |
| ID | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| WEB-001 | WEB | E-mail obrigatório | Automático | Cypress UI |
| WEB-002 | WEB | Senha obrigatória | Automático | Cypress UI |
| WEB-003 | WEB | E-mail sem @ | Automático | Cypress UI |
| WEB-004 | WEB | Senha curta | Automático | Cypress UI |
| WEB-005 | WEB | Feedback visual | Automático | Cypress UI |
| WEB-006 | WEB | Botão desabilitado | Automático | Cypress UI |
| WEB-007 | WEB | Loader exibido | Automático | Cypress UI |
| WEB-008 | WEB | Mostrar/ocultar senha | Automático | Cypress UI |
| WEB-009 | WEB | Redirecionamento após login | Automático | Cypress UI |
| WEB-010 | WEB | Logout | Automático | Cypress UI |
| WEB-011 | WEB | Sessão expirada | Semi-Automático | Cypress UI |

**CASOS DE TESTE BDD – WEB (Cypress UI)**

**WEB-001 – E-mail obrigatório**

**Cenário: Exibir mensagem para campo de e-mail obrigatório**

Dado que estou na página de Login do SGVC

Quando clico no botão “Entrar” sem preencher o campo de e-mail

Então o sistema deve exibir a mensagem "E-mail é obrigatório"

E deve manter o foco no campo de e-mail

**WEB-002 – Senha obrigatória**

**Cenário: Exibir mensagem para senha obrigatória**

Dado que estou na tela de Login

Quando tento realizar login preenchendo apenas o e-mail

E clico no botão “Entrar”

Então o sistema deve exibir “Senha é obrigatória”

**WEB-003 – E-mail sem @**

**Cenário: Exibir mensagem de erro ao inserir e-mail inválido**

Dado que estou na página de Login

Quando insiro o e-mail "usuario.com"

E clico em "Entrar"

Então o sistema deve exibir "Formato de e-mail inválido"

E o campo de e-mail deve ficar destacado visualmente

**WEB-004 – Senha curta**

**Cenário: Alertar senha com menos de 3 caracteres**

Dado que estou na página de Login

Quando insiro uma senha com apenas 2 caracteres

E clico em "Entrar"

Então deve ser exibida a mensagem "Senha muito curta"

**WEB-005 – Feedback visual de erro**

**Cenário: Exibir feedback visual em campos com erro**

Dado que estou na tela de Login

Quando envio credenciais inválidas

Então o sistema deve exibir uma mensagem geral "Credenciais inválidas"

E deve destacar os campos e-mail e senha com borda vermelha

**WEB-006 – Botão desabilitado durante envio**

**Cenário: Botão de login fica desabilitado durante autenticação**

Dado que estou na tela de Login

Quando preencho e-mail e senha válidos

E clico em “Entrar”

Então o botão deve ser desabilitado para evitar múltiplos envios

E deve permanecer assim até receber a resposta da API

**WEB-007 – Loader exibido**

**Cenário: Exibir loader enquanto a autenticação é processada**

Dado que estou na tela de Login

Quando envio o formulário de login

Então um indicador de carregamento (loader/spinner) deve ser exibido

E deve desaparecer após o retorno da API

**WEB-008 – Mostrar/ocultar senha**

**Cenário: Exibir senha quando usuário clicar no ícone de visualização**

Dado que estou na tela de Login

Quando clico no ícone de “mostrar senha”

Então o campo senha deve mudar para o tipo "text"

Quando clico novamente

Então o campo deve voltar para "password"

**WEB-009 – Redirecionamento após login**

**Cenário: Redirecionar usuário autenticado para o Dashboard**

Dado que envio credenciais válidas

Quando o login é bem-sucedido

Então devo ser redirecionado para /dashboard

E o token JWT deve ser armazenado localmente

**WEB-010 – Logout**

**Cenário: Usuário deve conseguir realizar logout**

Dado que estou autenticado no sistema

Quando clico na opção “Sair” ou “Logout”

Então devo ser redirecionado para a tela de Login

E o token salvo deve ser removido do armazenamento local

**WEB-011 – Sessão expirada**

**Cenário: Exibir mensagem ao tentar navegar com sessão expirada**

Dado que estou autenticado

E minha sessão expirou por tempo de inatividade

Quando tento acessar qualquer página interna do sistema

Então uma mensagem "Sessão expirada, faça login novamente" deve ser exibida

E devo ser redirecionado automaticamente para a página de Login

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABELA CONSOLIDADA DE TODOS OS CENÁRIOS- ACESSIBILIDADE – Axe / WCAG / NVDA** | | | | |
| **ID** | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| **A11Y-001** | Acessibilidade | Validação com Axe DevTools | Automático | Cypress + axe-core |
| **A11Y-002** | Acessibilidade | Leitor de tela real  Leitor | Manual | |  | | --- | | NVDA / VoiceOver |  |  | | --- | |  | |
| **A11Y-003** | Acessibilidade | Navegação por Tab | Manual | Teclado |
| **A11Y-004** | Acessibilidade | Contraste | Semi-automático | |  | | --- | | Axe DevTools |  |  | | --- | |  | |
| **A11Y-005** | Acessibilidade | |  | | --- | | Zoom 200% |  |  | | --- | |  | | Manual | WCAG |

**CASOS DE TESTE BDD – ACESSIBILIDADE (A11Y)**

**A11Y-001 — Validação com Axe DevTools**

**Cenário: Validar automaticamente violações de acessibilidade com Axe DevTools**

Dado que estou na página de Login do SGVC  
E o axe-core foi injetado na aplicação  
Quando executo a análise automática de acessibilidade  
Então o relatório do Axe não deve exibir violações críticas (severity: serious, critical)  
E elementos obrigatórios como rótulos, contrastes e landmarks devem estar corretamente definidos  
E o relatório deve estar acessível para auditoria

*(Este cenário será totalmente automatizado no Cypress.)*

**A11Y-002 — Leitor de tela real (NVDA / VoiceOver)**

**Cenário: Validar leitura correta do conteúdo por leitor de tela**

Dado que estou com NVDA ou VoiceOver ativado  
E estou na página de Login do SGVC  
Quando navego pelos elementos da página utilizando comandos do leitor de tela  
Então os textos devem ser lidos de forma clara e compreensível  
E todos os elementos interativos devem possuir descrições adequadas (aria-label, role, alt)  
E não devem existir botões ou inputs sem descrição acessível  
E a ordem de leitura deve seguir a ordem visual natural da página

*(Este cenário é sempre manual — leitores de tela não podem ser automatizados.)*

**A11Y-003 — Navegação completa via teclado (TAB / SHIFT+TAB)**

**Cenário: Validar navegação acessível usando apenas o teclado**

Dado que estou na tela de Login  
Quando navego entre os elementos usando as teclas TAB e SHIFT+TAB  
Então o foco deve seguir uma ordem lógica e previsível  
E todos os elementos interativos devem receber foco visível  
E nenhum componente deve “travar” o foco  
E o botão “Entrar” deve ser acionável via tecla ENTER ou SPACE

*(Manual — valida comportamento real de navegação.)*

**A11Y-004 — Verificação de contraste (WCAG AA)**

**Cenário: Validar contraste de cores conforme WCAG 2.1 nível AA**

Dado que estou na página de Login  
E executo uma análise de contraste com Axe DevTools  
Quando verifico elementos críticos como textos, labels, placeholders e botões  
Então o contraste entre texto e fundo deve ser no mínimo 4.5:1  
E elementos grandes devem ter contraste mínimo de 3:1  
E nenhum aviso de contraste insuficiente deve aparecer no relatório

*(Semi-automático — parte automatizável via Axe, parte visual.)*

**A11Y-005 — Zoom 200% (WCAG)**

**Cenário: Validar a usabilidade da página com zoom de 200%**

Dado que estou na página de Login do SGVC  
E aplico zoom de 200% no navegador  
Quando visualizo a interface  
Então todos os elementos devem permanecer legíveis  
E não deve ocorrer quebra de layout  
E nenhum elemento deve sair da área de visualização  
E o usuário deve conseguir se autenticar normalmente com o zoom aplicado

*(Manual — requer inspeção visual e comportamento real.)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABELA CONSOLIDADA DE TODOS OS CENÁRIOS-** **RESPONSIVIDADE – Cypress viewport** | | | | |
| **ID** | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| **RSP-001** | Responsividade | Mobile 360px | Automático | Cypress viewport |
| **RSP-002** | Responsividade | Tablet 768px | Automático | Cypress viewport |
| **RSP-003** | Responsividade | Desktop 1920px | Automático | Cypress viewport |
| **RSP-004** | Responsividade | Landscape | Automático | Cypress viewport |
| **RSP-005** | Responsividade | Quebra de layout | Manual | Cypress viewport rotate |

**CASOS DE TESTE BDD – RESPONSIVIDADE (RSP)**

**RSP-001 — Mobile 360px**

**Cenário: Validar layout da página de Login em resolução Mobile 360px**

Dado que estou na página de Login do SGVC  
Quando defino a resolução da tela para **360x640** (modo mobile)  
Então todos os elementos devem ser exibidos dentro da viewport  
E o botão “Entrar” deve ficar visível sem necessidade de scroll horizontal  
E não deve existir barra de rolagem lateral  
E o formulário deve permanecer centralizado e legível  
E o layout deve manter boa usabilidade em toque (touch-friendly)

**RSP-002 — Tablet 768px**

**Cenário: Validar layout da página de Login em resolução Tablet 768px**

Dado que estou na tela de Login  
Quando defino a resolução para **768x1024**  
Então o layout deve adaptar-se para o modo tablet  
E o formulário deve permanecer alinhado e sem distorções  
E os elementos não devem se sobrepor  
E não deve haver quebra de texto ou desalinhamento de botões

**RSP-003 — Desktop 1920px**

**Cenário: Validar layout em resolução Desktop FullHD**

Dado que estou na tela de Login do SGVC  
Quando defino a resolução para **1920x1080**  
Então o layout deve preencher o espaço disponível de forma equilibrada  
E o formulário não deve parecer desproporcionalmente pequeno  
E nenhum elemento deve “esticar” fora do espaço esperado  
E a interface deve manter alinhamento visual consistente

**RSP-004 — Landscape**

**Cenário: Validar layout em modo landscape**

Dado que estou na tela de Login  
Quando simulo o dispositivo em modo **landscape**  
Então os componentes devem reorganizar-se sem quebrar  
E o formulário deve permanecer legível  
E o botão de login deve continuar acessível  
E a interface não deve apresentar overflow lateral

**RSP-005 — Quebra de layout (Manual)**

**Cenário: Identificar possíveis quebras de layout em múltiplas resoluções**

Dado que estou testando o SGVC em resoluções variadas  
E utilizo as viewports mobile, tablet e desktop  
Quando observo o comportamento visual da interface  
Então nenhum elemento deve sair da área visível  
E não deve ocorrer sobreposição de texto, botões ou inputs  
E a interface deve permanecer utilizável em todas as resoluções  
E qualquer quebra deve ser registrada com evidência visual (screenshot)

*(Este cenário é manual porque depende da inspeção humana.)*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABELA CONSOLIDADA DE TODOS OS CENÁRIOS-** **SEGURANÇA – Postman / Scripts / SuperTest** | | | | |
| **ID** | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| **SEC-001** | Segurança | SQL Injection | Automático | Postman / SuperTest |
| **SEC-002** | Segurança | XSS | Automático | Postman |
| **SEC-003** | Segurança | Brute Force | Automático | Scripts + Postman |
| **SEC-004** | Segurança | Replay Token | Automático | Postman |
| **SEC-005** | Segurança | Token expirado | Automático | SuperTest |
| **SEC-006** | Segurança | Sessão expirada | Semi-automático | SuperTest |

**CASOS DE TESTE BDD – SEGURANÇA (SEC)**

**SEC-001 — SQL Injection**

**Cenário: Prevenir SQL Injection no campo de e-mail e senha**

Dado que estou testando o endpoint /auth/login  
E envio payloads maliciosos contendo parâmetros de SQL Injection como:  
| ' OR '1'='1, "admin' --", "1 OR 1=1" |  
Quando envio uma requisição POST para /auth/login com dados maliciosos  
Então a API deve retornar status **401** ou **400**  
E não deve autenticar o usuário  
E não deve vazar erro de banco de dados  
E o corpo da resposta não deve conter stacktrace, queries ou mensagens sensíveis  
E a tentativa deve ser registrada como suspeita (se log estiver habilitado)

**SEC-002 — XSS (Cross-Site Scripting)**

**Cenário: Impedir execução de scripts maliciosos no campo de e-mail**

Dado que estou testando payload XSS  
E envio no campo de e-mail valores como "<script>alert('xss')</script>"  
Quando envio POST para /auth/login  
Então a API deve retornar erro 400  
E deve sanitizar ou rejeitar qualquer elemento que contenha scripts  
E não deve refletir o script na resposta  
E não deve permitir que o script seja executado no navegador

**SEC-003 — Brute Force Attack**

**Cenário: Impedir ataques de força bruta com múltiplas tentativas**

Dado que um atacante tenta realizar várias tentativas de login  
E executo mais de 10 tentativas consecutivas com senhas incorretas  
Quando executo um script automatizado enviando tentativas rápidas  
Então a API deve bloquear temporariamente o IP ou o usuário  
E deve responder com código 429 (Too Many Requests) ou 403 (Forbidden)  
E deve exibir mensagem "Muitas tentativas, tente novamente mais tarde"  
E não deve permitir login mesmo com credenciais corretas durante o bloqueio

**SEC-004 — Replay Token**

**Cenário: Impedir reutilização de tokens já utilizados (Replay Attack)**

Dado que obtenho um token JWT válido após login  
E utilizo esse token em uma requisição autorizada  
Quando tento reutilizar o mesmo token imediatamente após expirado ou revogado  
Então a API deve retornar **401**  
E deve exibir "Token inválido ou já utilizado"  
E não deve permitir que requisições antigas sejam reaplicadas  
E tokens antigos não devem funcionar após logout

**SEC-005 — Token expirado**

**Cenário: Impedir acesso com token JWT expirado**

Dado que possuo um token JWT cujo tempo de expiração já passou  
Quando envio o token expirado no header Authorization  
E tento acessar /clientes, /produtos ou qualquer rota protegida  
Então a API deve retornar status **401**  
E deve exibir "Token expirado"  
E deve exigir novo login para continuar

**SEC-006 — Sessão expirada**

**Cenário: Sessão expira por inatividade prolongada**

Dado que estou autenticado no sistema  
E permaneço inativo por mais de 30 minutos  
Quando envio uma nova requisição a qualquer rota protegida  
Então a API deve retornar status **401**  
E deve exibir "Sessão expirada, faça login novamente"  
E não deve permitir acesso até um novo login ser realizado

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABELA CONSOLIDADA DE TODOS OS CENÁRIOS– K6** | | | | |
| **ID** | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| **PERF-001** | Performance | Login – 100 usuários simultâneos | Automático | K6 |
| **PERF-002** | Performance | Registrar 1000 vendas | Automático | K6 |
| **PERF-003** | Performance | Latência P95 | Automático | K6 |
| **PERF-004** | Performance | Teste de carga constante | Automático | K6 |
| **PERF-005** | Performance | Teste de stress | Automático | K6 |

**CASOS DE TESTE BDD – PERFORMANCE (K6)**

**PERF-001 — Login com 100 usuários simultâneos**

**Cenário: Validar estabilidade do endpoint de login com 100 usuários simultâneos**

Dado que tenho um script K6 configurado para simular 100 usuários virtuais  
E todos tentam realizar login no mesmo endpoint /auth/login  
Quando executo o teste de carga simultânea  
Então ao menos 95% das requisições devem responder com status 200  
E a taxa de erro deve ser inferior a 1%  
E nenhum erro 500 ou 503 deve ocorrer  
E o tempo de resposta P95 deve ser inferior a 100ms

**PERF-002 — Registrar 1000 vendas consecutivas**

**Cenário: Validar estabilidade ao registrar 1000 vendas contínuas**

Dado que tenho um script K6 configurado para enviar múltiplas requisições POST para /vendas  
E cada requisição registra uma venda válida  
Quando executo o teste até completar 1000 vendas  
Então a API não deve apresentar erros 500  
E o consumo de CPU e memória deve permanecer estável  
E o tempo de resposta médio deve ficar dentro do esperado  
E nenhuma venda deve ser registrada com dados inconsistentes

**PERF-003 — Latência P95**

**Cenário: Garantir que a latência P95 esteja dentro do limite aceitável**

Dado que estou executando um teste de performance variando o número de usuários  
E monitoro a latência das respostas  
Quando o K6 calcula a latência percentil 95 (P95)  
Então o valor deve ser menor que 100ms  
E nenhuma rota crítica deve ultrapassar esse limite  
E um relatório deve ser gerado para auditoria

**PERF-004 — Teste de carga constante**

**Cenário: Validar comportamento da API sob carga constante por 5 minutos**

Dado que configurei o K6 para simular usuários constantes ao longo de 5 minutos  
Quando executo o teste no endpoint /auth/login e /vendas  
Então o sistema deve manter estabilidade sem aumento brusco de latência  
E a taxa de requisições por segundo (RPS) deve permanecer constante  
E o erro médio deve ser inferior a 1%  
E nenhum serviço deve cair durante a execução

**PERF-005 — Teste de stress**

**Cenário: Avaliar o comportamento da API sob carga crescente até o limite máximo**

Dado que configurei o K6 para aumentar gradualmente o número de usuários  
E atingir um volume acima da capacidade normal da API  
Quando executo o teste de stress  
Então o sistema deve identificar o ponto de saturação  
E não deve corromper dados mesmo após falhas  
E a API deve recuperar o comportamento normal após o teste  
E o relatório deve mostrar claramente o momento em que o sistema degradou

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABELA CONSOLIDADA DE TODOS OS CENÁRIOS-** **MANUAL – Exploratório** | | | | |
| **ID** | **Categoria** | **Cenário** | **Automação** | **Ferramenta** |
| **MAN-001** | Manual | UX real | Manual | Exploratório |
| **MAN-002** | Manual | Leitor de tela real | Manual | NVDA / Jaws |
| **MAN-003** | Manual | Contraste real | Manual | Teste humano |
| **MAN-004** | Manual | Falha real de rede | Manual | Ambiente |
| **MAN-005** | Manual | API realmente caindo | Manual | |  | | --- | | Ambiente instável |  |  | | --- | |  | |
| **MAN-006** | Manual | Comportamento humano inesperado | Manual | Exploratório |

**CASOS DE TESTE BDD — MANUAL / EXPLORATÓRIO**

**MAN-001 — UX real**

**Cenário: Avaliar experiência real do usuário na interface de Login**

Dado que estou acessando o sistema como um usuário real  
Quando interajo com os campos de e-mail, senha e botão de login  
Então a interface deve apresentar clareza visual  
E os textos devem ser legíveis em todos os elementos  
E os botões devem ter tamanho adequado para clique ou toque  
E a jornada de login deve ser intuitiva e sem esforço  
E nenhum comportamento confuso ou inesperado deve ocorrer

**MAN-002 — Leitor de tela real**

**Cenário: Validar leitura correta da interface por leitores de tela reais**

Dado que estou utilizando o leitor de tela NVDA ou JAWS  
Quando navego pelos elementos da tela de login  
Então todos os campos devem ser anunciados corretamente  
E botões devem conter rótulos claros (aria-label, roles, alt-text)  
E mensagens de erro devem ser lidas automaticamente  
E a ordem de leitura deve seguir a organização visual da página

**MAN-003 — Contraste real**

**Cenário: Validar contraste real com percepção humana**

Dado que estou avaliando a tela de login em monitor real  
Quando observo textos, botões, placeholders e avisos  
Então os elementos devem ter contraste visual confortável  
E devem estar legíveis a diferentes distâncias  
E não devem desaparecer em ambientes com mais ou menos luminosidade  
E nenhuma cor deve comprometer a leitura ou a distinção de componentes

*(Esse teste é manual porque percepção visual varia entre usuários e ambientes.)*

**MAN-004 — Falha real de rede**

**Cenário: Validar comportamento da UI diante de queda real de conexão**

Dado que estou conectado ao SGVC  
E minha conexão cai ou oscila de forma real  
Quando tento realizar login ou mudar de página  
Então o sistema deve exibir mensagem amigável de erro  
E não deve travar a interface  
E deve permitir nova tentativa após reconexão  
E não deve perder o estado de forma inesperada

*(Esse teste depende de redes reais, não é possível simular tudo por automação.)*

**MAN-005 — API realmente caindo**

**Cenário: Validar comportamento do sistema quando a API está realmente indisponível**

Dado que a API do SGVC está fora do ar ou instável  
Quando o usuário tenta realizar operações como login ou carregamento de dados  
Então o sistema deve exibir mensagens amigáveis como "Serviço indisponível"  
E não deve exibir erros técnicos como stacktrace  
E deve evitar que a interface quebre ou fique congelada  
E deve orientar o usuário a tentar novamente mais tarde

*(Depende de ambiente real — instabilidade não é totalmente simulável)*

**MAN-006 — Comportamento humano inesperado**

**Cenário: Explorar interações imprevisíveis do usuário**

Dado que um usuário pode realizar ações fora do fluxo esperado  
Quando o usuário tenta, por exemplo:

* digitar textos aleatórios
* usar emojis no login
* pressionar teclas repetidamente
* alternar de aba rapidamente
* clicar várias vezes no mesmo botão  
  Então a UI deve responder de forma estável  
  E não deve travar  
  E não deve apresentar falhas gráficas  
  E não deve duplicar ações ou gerar estados inconsistentes