**Interface gráfica do usuário, Texto, Site

O conteúdo gerado por IA pode estar incorreto.**

**Plano e Estratégia**

**de Testes Adaptada**

Baseado na ISO-29119-3.

**1. Épico e Estimativa Geral de Esforço em Testes**

**Link:**[**Épico – Gestão de Empreendedores e Faturamento**](https://github.com/jeff-barbosa123/Software-para-Gest-o-de-Vendas-e-Clientes/issues/1)

Esforço: Alta

**2. User Stories e Estimativa de Esforço em Testes**

|  |  |  |
| --- | --- | --- |
| Código | Descrição | Esforço |
| User Story-001–SGVC | Login de Empreendedores | Médio |
| User Story 002 – SGVC | Cadastro de Clientes e Produtos | Médio |
| User Story 003 – SGVC | Registro de Vendas e Faturamento | Alto |
| User Story 004 – SGVC | Visualização do Total de Faturamento | Alto |

**3. Condições de Teste e Camadas**

**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)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Condição | Resultado Esperado | Camada |
| 1 | Validar obrigatoriedade de campos de login | Sistema deve solicitar preenchimento antes do envio | Web |
| 2 | Validar autenticação com credenciais válidas e inválidas | Acesso permitido ou negado conforme credenciais | API |
| 3 | Validar bloqueio após múltiplas tentativas falhas | Acesso bloqueado temporariamente | API |
| 4 | Validar expiração de sessão por inatividade | |  | | --- | | Sessão encerrada após tempo limite |  |  | | --- | |  | | Web |
| 5 | Validar mensagens de erro e feedback ao usuário | |  | | --- | | feedback ao usuário |  |  | | --- | | Mensagens apresentadas de forma clara | | API |

**User Story 002 – SGVC: Cadastro de Clientes e Produtos**

**Link:** [User Story 002 – SGVC: Cadastro de Clientes e Produtos](User%20Story%20002%20–%20SGVC:%20Cadastro%20de%20Clientes%20e%20Produtos)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Condição | Resultado Esperado | Camada |
| 1 | Validar obrigatoriedade de campos de cadastro | Sistema impede cadastro incompleto | Web |
| 2 | |  | | --- | | Validar duplicidade de dados |  |  | | --- | |  | | Cadastro rejeitado quando duplicado | API |
| 3 | Validar edição e exclusão de registros | Operações refletidas corretamente no sistema | API |
| 4 | Validar consistência e formato dos dados | Dados armazenados em formato válido | API |
| 5 | |  | | --- | | Validar rejeição de valores inválidos |  |  | | --- | |  | | |  | | --- | | Sistema retorna erro de validação |  |  | | --- | |  | | Web |

**User Story 00**3 **– SGVC: Registro de Vendas e Faturamento**

**Link:** [**User Story 003 – SGVC: Registro de Vendas e Faturamento**](https://github.com/jeff-barbosa123/Software-para-Gest-o-de-Vendas-e-Clientes/issues/4)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Condição | Resultado Esperado | Camada |
| 1 | Validar obrigatoriedade dos campos de venda | Sistema impede registro incompleto | Web |
| 2 | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Validar vínculo entre cliente, produto e venda | |  |  | | --- | |  | | |  | | --- | | Venda associada corretamente |  |  | | --- | |  | | API |
| 3 | Validar cálculo automático de faturamento | Totais atualizados conforme vendas | API |
| 4 | |  | | --- | |  |  |  | | --- | | Validar edição e cancelamento de venda | | Alterações refletidas corretamente | API |
| 5 | |  |  |  | | --- | --- | --- | | |  | | --- | | Validar integridade de valores e datas |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Sistema rejeita dados inválidos |  |  | | --- | |  | | Web |

**User Story 004: Registro de Vendas e Faturamento**

**Link:** [**User Story 004 – SGVC: Visualização do Total de Faturamento**](https://github.com/jeff-barbosa123/Software-para-Gest-o-de-Vendas-e-Clientes/issues/5)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Condição | Resultado Esperado | Camada |
| 1 | Validar exibição dos totais por período e status | Totais exibidos corretamente conforme filtros | API |
| 2 | |  |  |  | | --- | --- | --- | | |  | | --- | |  |  |  | | --- | | Validar funcionamento dos filtros disponíveis | |  |  | | --- | |  | | |  | | --- | | Dados apresentados conforme critérios |  |  | | --- | |  | | API |
| 3 | |  | | --- | |  |  |  | | --- | | Validar consistência entre listagem e totais | | Informações exibidas de forma coerente | API |
| 4 | |  | | --- | |  |  |  |  |  | | --- | --- | --- | | |  | | --- | | Validar exportação de relatórios |  |  | | --- | |  | | | Relatórios gerados com dados corretos | Web |
| 5 | |  |  |  | | --- | --- | --- | | |  | | --- | | Validar comportamento quando não há dados |  |  | | --- | |  | |  |  | | --- | |  | | |  | | --- | | Sistema informa ausência de registros |  |  | | --- | |  | | Web |

**4. Missões de Teste Exploratório**

**Login de Empreendedores**

1. Explore a funcionalidade Login de Empreendedores, com as heurísticas **Strings longas e Caracteres especiais**, para descobrir se o endpoint valida e normaliza corretamente e-mails e senhas com 255+ caracteres.
2. Explore a funcionalidade Login de Empreendedores, com a heurística **Injeção de SQL**, para descobrir se entradas maliciosas são neutralizadas e registradas sem vazamento de informações.
3. Explore a funcionalidade Login de Empreendedores, com as heurísticas **Falhas agrupadas e Abordagem contrária**, para descobrir se múltiplos erros de login (ex.: senha incorreta 3x) resultam em bloqueio temporário.
4. Explore a funcionalidade Login de Empreendedores, com as heurísticas **Em branco e Espaços extras**, para descobrir se o backend realiza trim() e rejeita credenciais vazias.
5. Explore a funcionalidade Login de Empreendedores, com as heurísticas **Interrupções e Timeout**, para descobrir se o fluxo de autenticação mantém o estado durante perda de conexão.
6. Explore a funcionalidade Login de Empreendedores, com a heurística **Tempo limite**, para descobrir se a sessão expira corretamente após 30 minutos de inatividade.
7. Explore a funcionalidade Login de Empreendedores, com as heurísticas **Caracteres acentuados e asiáticos**, para descobrir se o sistema aceita entradas internacionais e mantém codificação UTF-8.

**Cadastro de Clientes**

1. Explore a funcionalidade Cadastro de Clientes, com as heurísticas **CRUD e Siga os Dados**, para descobrir se a integridade se mantém entre cliente e registros de venda.
2. Explore a funcionalidade Cadastro de Clientes, com as heurísticas **Ataques de Tipos de Dados e Formatos de data**, para descobrir se o endpoint rejeita datas inválidas e converte corretamente formatos incorretos.
3. Explore a funcionalidade Cadastro de Clientes, com a heurística **Nome já existente**, para descobrir se há validação de duplicidade e retorno de erro HTTP 409 (Conflict).
4. Explore a funcionalidade Cadastro de Clientes, com as heurísticas **Em branco e Strings longas**, para descobrir se campos textuais possuem limites e validações coerentes.
5. Explore a funcionalidade Cadastro de Clientes, com a heurística **Protegido contra gravação**, para descobrir se clientes “inativos” não podem ser alterados.
6. Explore a funcionalidade Cadastro de Clientes, com as heurísticas **Interrupções e Requisições repetidas**, para descobrir se o sistema evita duplicidade de cadastro após falha de rede.
7. Explore a funcionalidade Cadastro de Clientes, com a heurística **Diferença de horários entre máquinas**, para descobrir se o cadastro é consistente em diferentes fusos horários.

**Cadastro de Produtos**

1. Explore a funcionalidade Cadastro de Produtos, com as heurísticas **Ataques de Tipos de Dados e Strings longas**, para descobrir se o sistema trata nomes e códigos com tamanhos extremos.
2. Explore a funcionalidade Cadastro de Produtos, com a heurística **Abordagem contrária**, para descobrir se o sistema impede valores negativos de preço e estoque.
3. Explore a funcionalidade Cadastro de Produtos, com as heurísticas **Siga os Dados e Testes como Experimentos**, para descobrir se a criação de produtos impacta corretamente os relatórios de venda e faturamento.
4. Explore a funcionalidade Cadastro de Produtos, com as heurísticas **Falhas agrupadas e CRUD**, para descobrir se erros em uma operação (PUT, DELETE) afetam outras rotas.
5. Explore a funcionalidade Cadastro de Produtos, com a heurística **Bloqueado e Indisponível**, para descobrir se produtos bloqueados são impedidos de serem vendidos.
6. Explore a funcionalidade Cadastro de Produtos, com a heurística **Corrompido**, para descobrir se o sistema reage bem a imagens de produto inválidas ou arquivos corrompidos.
7. Explore a funcionalidade Cadastro de Produtos, com a heurística **Tempo limite**, para descobrir se o salvamento de produtos grandes falha de forma segura em caso de lentidão.

**Registro de Vendas e Faturamento**

1. Explore a funcionalidade Registro de Vendas, com as heurísticas **Siga os Dados e Interrupções**, para descobrir se há rollback automático em caso de falha durante o registro da venda.
2. Explore a funcionalidade Registro de Vendas, com as heurísticas **Ataques de Tipos de Dados e Valores extremos**, para descobrir se o backend rejeita valores negativos ou exageradamente altos.
3. Explore a funcionalidade Registro de Vendas, com as heurísticas **CRUD e Falhas agrupadas**, para descobrir se atualizações em uma venda refletem corretamente no total de faturamento.
4. Explore a funcionalidade Registro de Vendas, com as heurísticas **Tempo limite e Duplicações**, para descobrir se o endpoint lida corretamente com reenvios automáticos de requisição.
5. Explore a funcionalidade Registro de Vendas, com a heurística **Em branco e Siga os Dados**, para descobrir se vendas sem cliente associado são corretamente bloqueadas.
6. Explore a funcionalidade Registro de Vendas, com a heurística **Diferença de horários entre máquinas**, para descobrir se o faturamento é coerente em diferentes regiões.
7. Explore a funcionalidade Registro de Vendas, com a heurística **Testes como Experimentos**, para descobrir se o sistema arredonda corretamente valores de centavos ao calcular total e imposto.

**Visualização do Total de Faturamento**

1. Explore a funcionalidade Visualização do Total de Faturamento, com as heurísticas **CRUD e Siga os Dados**, para descobrir se o cálculo total é atualizado em tempo real após cada venda.
2. Explore a funcionalidade Visualização do Total de Faturamento, com as heurísticas **Tempo limite e Interrupções**, para descobrir se o gráfico se recupera após falhas da API.
3. Explore a funcionalidade Visualização do Total de Faturamento, com as heurísticas **Abordagem contrária e Injeção de SQL**, para descobrir se o endpoint /faturamento/total rejeita parâmetros manipulados na URL.
4. Explore a funcionalidade Visualização do Total de Faturamento, com as heurísticas **Falhas agrupadas e Testes como Experimentos**, para descobrir se inconsistências nos registros de venda afetam o total apresentado.
5. Explore a funcionalidade Visualização do Total de Faturamento, com as heurísticas **Diferença de horários entre máquinas**, para descobrir se há divergência no cálculo diário em sistemas com clocks desincronizados.
6. Explore a funcionalidade Visualização do Total de Faturamento, com a heurística **Interface e Usabilidade**, para descobrir se o layout do dashboard mantém legibilidade e coerência após múltiplas atualizações.
7. Explore a funcionalidade Visualização do Total de Faturamento, com as heurísticas **Ataques de Tipos de Dados e Resiliência**, para descobrir se o front-end lida corretamente com respostas nulas, JSON malformado ou erro 500.

**5. Testes Não-Funcionais**

|  |  |  |
| --- | --- | --- |
| Tipo | Teste | Resultado Esperado |
| Performance | Avaliar se as operações de **login, cadastro de clientes e registro de vendas** respondem em tempo adequado com **100 usuários simultâneos**. | Percentil de 95 < **100 ms** |
| Carga | Testar a estabilidade do sistema ao registrar **1000 vendas consecutivas** via API, verificando uso de CPU e memória. | Nenhum erro 5xx e consumo de recursos abaixo de 80% |
| |  | | --- | | **Segurança** |  |  | | --- | |  | | Avaliar endpoints sensíveis (/auth/login, /clientes, /faturamento/total) contra **injeção SQL, XSS e brute force**. | Nenhuma vulnerabilidade detectada |
| Confiabilidade | Verificar se o sistema mantém a **integridade dos dados** após falha de rede ou reinício durante o registro de vendas. | Dados restaurados e consistentes |
| Usabilidade | Avaliar se o **tempo de carregamento e atualização do painel de faturamento** é inferior a **2 segundos** e mantém boa legibilidade. | Interface fluida e responsiva |

**6. Automação de Testes**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Condição | Resultado Esperado | Camada |
| 1 | Realizar **login** com credenciais válidas (admin@negocio.com / admin123) | Sistema autentica o usuário e retorna token JWT válido. | API |
| 2 | Tentar login com **senha incorreta 3x seguidas** | Acesso bloqueado temporariamente (HTTP 403) por 15 minutos. | API |
| 3 | |  | | --- | |  |  |  | | --- | | Tentar login com campo de e-mail vazio | | Sistema exibe mensagem de erro de campo obrigatório. | Web |
| 4 | Cadastrar novo cliente com e-mail único e campos válidos | Cliente criado com sucesso (HTTP 201). | API |
| 5 | Tentar cadastrar cliente com e-mail duplicado | Sistema retorna erro de conflito (HTTP 409). | API |
| 6 | Cadastrar novo produto com nome e preço válidos | Produto criado com sucesso e disponível para venda. | API |
| 7 | Tentar cadastrar produto com preço negativo | Sistema rejeita a operação (HTTP 422 – violação de regra de negócio). | API |
| 8 | Registrar nova venda com cliente e produto vinculados | Venda gravada e total de faturamento atualizado corretamente. | API |
| 9 | Tentar registrar venda sem cliente ou produto vinculado | Sistema rejeita operação (HTTP 422). | API |
| 10 | Cancelar venda registrada | Venda marcada como cancelada e excluída do cálculo de faturamento. | API |
| 11 | Consultar relatório de faturamento diário | Retorna dados consolidados no formato JSON. | API |
| 12 | Exportar relatório de faturamento em PDF | Gera e baixa o arquivo PDF corretamente. | API e Web |

**7. Mapeamento dos Dados de Teste**

Dado em Banco, Configuração ou Mock

|  |  |  |  |
| --- | --- | --- | --- |
| Dado | Tipo | Responsável | Status |
| Usuário padrão de autenticação (admin@negocio.com / admin123) | Dado em Banco (demo) | Equipe QA | Criado |
| Token JWT válido (retornado via /auth/login) | Configuração de Ambiente (.env) | Jefferson Paulo | Gerado |
| Cliente de teste: joao@cliente.com, nome “João Cliente” | Mock (POST /customers) | Jefferson Paulo | Criado |
| Cliente duplicado: joao@cliente.com | Mock (POST /customers) | Jefferson Paulo | Criado (para teste 409) |
| Produto de teste: “Bolo Caseiro” – preço R$ 20,00 | Dado em Banco (POST /products) | Jefferson Paulo | Criado |
| Produto inválido: “Produto Teste” – preço -10 | Mock (POST /products) | Jefferson Paulo | Criado (para teste 422) |
| Venda válida (Cliente João, Produto Bolo Caseiro, valor R$ 20,00) | Mock (POST /sales) | |  | | --- | | Equipe QA |  |  | | --- | |  | | Criado |
| Venda inválida (sem cliente vinculado) | Mock (POST /sales) | |  | | --- | | Equipe QA |  |  | | --- | |  | | Criado (para teste 422) |
| Relatório de faturamento diário (/reports/revenue) | |  | | --- | | Geração via API |  |  | | --- | |  | | |  | | --- | |  |  |  | | --- | | Equipe QA | | Em execução |
| Relatório PDF (/reports/revenue/export?format=pdf) | Exportação | |  | | --- | |  |  |  | | --- | | Equipe QA | | Em teste |

8. Defeitos Conhecidos

Status: levantamento parcial — defeitos observados nas execuções iniciais de testes funcionais e exploratórios.  
Novos achados serão adicionados conforme o avanço das execuções automatizadas.

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Defeito | Camada | Status |
| 1 | **Ao realizar um teste com cadastrar telefone de formato invalido por Exemplo:123456** | **API** | **🟥 Aberto** |
| 2 | **Endpoint /reports/revenue às vezes exibe valores desatualizados após cancelar vendas.** | **API** | **🟨 Em observação** |
| 3 | **Campos de login (email, senha) não exibem mensagem clara para campos vazios.** | **Web** | **🟨 Em análise** |
| 4 | **Sessão JWT expira antes dos 30 minutos configurados em alguns testes consecutivos.** | **API** | **🟥 Aberto** |

Legenda de Status

🟥 Aberto – defeito confirmado e pendente de correção.  
🟨 Em observação – comportamento intermitente ou não confirmado.  
🟩 Fechado – validado como corrigido.

**Apoio de Inteligência Artificial**

Este plano foi desenvolvido com apoio de Inteligência Artificial (IA) para organizar, estruturar e otimizar a documentação de QA, mantendo alinhamento com os padrões da ISO 29119-3 e boas práticas ágeis.

**A IA foi utilizada para:**

Estruturar tabelas, formatações e seções do documento;

Acelerar a organização das User Stories, condições e missões de teste;

Garantir padronização visual e coerência terminológica em todo o plano.

**🧠 Validação humana:**

Todo o conteúdo técnico, análises de risco, definições de testes e evidências foram revisados e validados manualmente por Jefferson Paulo de Aguiar Barbosa (QA Responsável), assegurando qualidade, rastreabilidade e confiabilidade dos resultados.

📌 O uso ético e estratégico da IA faz parte da cultura de inovação aplicada ao QA moderno, unindo produtividade e precisão técnica.