

Qualidade e Automação em Plataforma de Vendas Online

Contexto Geral

O produto é uma plataforma de vendas online que conecta vendedores locais a consumidores, permitindo a criação de catálogos, adição de produtos ao carrinho e finalização de compras. A engenharia já possui esteiras de CI/CD em funcionamento, mas os testes automatizados ainda não cobrem os principais fluxos de ponta a ponta. Atualmente, falhas em produção têm gerado retrabalho e perda de confiança na entrega.

1. Diagnóstico e Priorização

Fluxos críticos para automação inicial:

- Cadastro de usuário (criação de conta).
- Login com credenciais válidas.
- Adição de produtos ao carrinho.
- Checkout (endereço, transporte e pagamento).
- Validação do pedido confirmado.

Critérios de priorização:

- Impacto direto no fluxo de compra (experiência do usuário).
- Frequência de uso (funcionalidades acessadas por todos os clientes).
- Risco de falha em produção.
- Cobertura de ponta a ponta (E2E) para garantir confiabilidade.

2. Estratégia Técnica

Tipos de testes adequados neste primeiro momento:

- Testes End-to-End (E2E): validação dos fluxos críticos de compra.
- Testes de API: garantir integração entre frontend e backend.
- Testes de Integração: verificar comunicação entre módulos (ex: carrinho ↔ pagamento).

Ferramentas e frameworks:

- Cypress: automação de testes E2E no frontend, com boa integração a CI/CD.
- Faker.js: geração de massa de dados dinâmica.
- Mochawesome: relatórios claros e detalhados para stakeholders.
- Cypress Real Events / Iframe: simulação real de interações e suporte a iframes.

Manutenibilidade e escalabilidade:

- Uso de Page Object Model (POM) para centralizar seletores.
- Criação de Custom Commands para ações repetitivas (ex: login).
- Estrutura modular (testes separados por tela/fluxo).
- Geração automática de relatórios para acompanhamento contínuo.

3. Implementação (Prova de Conceito)

Foi utilizado o site de demonstração <http://www.automationpractice.pl/index.php> como base para implementação. O fluxo automatizado cobre cadastro, login, adição de produto ao carrinho e checkout.

O código foi estruturado utilizando boas práticas de POM, fixtures para massa de dados e comandos customizados. Um README.md documenta os passos para instalação, execução e geração de relatórios.

Repositório: https://github.com/brunogsiq/QA_Proj_Cypress_AutomationPractice

3.1 Principais fluxo para automação

00 – Smoke Test (Geral)

- **Propósito:** checagem rápida de “sistema respirando” em cada PR/CI.
- **Valor:** detecta quebras grosseiras (rota off, assets críticos, layout/baseUrl), evitando gastar minutos/horas em suites longas quando o básico já falhou.
- **Métrica-alvo:** < 3–5 min, pass rate \geq 99%.

01 – Tela Home

- **Propósito:** porta de entrada do funil (navegação, menu, busca, banners).
- **Valor:** garante que o usuário consegue iniciar a jornada; se a Home quebra, **100% da receita é impactada**.
- **Riscos cobertos:** links de navegação, categorias, disponibilidade do catálogo inicial.

02 – Tela Cadastro

- **Propósito:** criar conta (necessária para pedidos e histórico).
- **Valor:** **desbloqueia retenção e personalização**; sem cadastro, muitos fluxos de pós-venda e comunicação falham.
- **Riscos cobertos:** validação de email/senha, obrigatórios, regras de país/CEP, mensagens de erro claras.

03 – Tela Login

- **Propósito:** autenticar cliente recorrente.
- **Valor:** login frágil derruba **conversão de retorno**, acesso a endereços salvos e checkout acelerado. É o caminho mais comum para quem já comprou.
- **Riscos cobertos:** credenciais válidas/inválidas, bloqueios, navegação pós-login (header “My account”), logout.

04 – Your Address

- **Propósito:** cadastro/gestão de endereço de entrega/cobrança.
- **Valor:** etapa crítica do **checkout** (cálculo de frete, impostos, SLA). Sem endereço válido, não existe pedido.
- **Riscos cobertos:** obrigatoriedade de telefone, CEP/estado/país, alias, consistência entre **Delivery** e **Billing**.
- **Efeito colateral positivo:** diminui chamados de suporte por falhas em endereço.

05 – Cart (Carrinho)

- **Propósito:** seleção final de itens (quantidade, remover, totais) e início do checkout.
- **Valor:** **diretamente ligado à receita** — qualquer bug aqui impacta conversão imediata.
- **Riscos cobertos:** modal “adicionado com sucesso”, resumo do carrinho, atualização de quantidade, totais, transição para **Address** → **Shipping** → **Payment**.
- **Ganho adicional:** cria base para testes de promoção/cupom, estoque e cálculo de frete.

Por que esse conjunto é o “mínimo viável” de alta cobertura?

- **Cobre o funil inteiro:** Home → Cadastro/Login → Endereço → Carrinho → Checkout (até Payment).
- **Maximiza detecção precoce de regressões:** qualquer quebra crítica nos pontos acima derruba conversão.
- **Escalável:** a partir desses fluxos, você pluga variações (busca, filtros, múltiplas categorias, métodos de frete/pagamento).

4. Entrega de Valor

Primeiros 30 dias:

- Mapear e priorizar fluxos críticos.
- Criar base inicial de testes automatizados (POC).
- Garantir integração na pipeline de CI/CD.
- Gerar relatórios automáticos para o time.

Equilíbrio entre entregas rápidas e base sustentável:

- Entregar valor imediato com automação de um fluxo crítico (checkout).
- Estruturar a base do projeto (Pages, Commands, Fixtures).
- Expandir gradualmente cobertura sem comprometer qualidade.

5. Colaboração e Cultura

Ações para cultura de qualidade:

- Workshops com devs e PO sobre boas práticas de testes.
- Revisão de PRs com foco em cobertura de testes.
- Dashboard de resultados dos testes para transparência.
- Incentivar criação de testes unitários e integração desde o desenvolvimento.

Como lidar com baixa cobertura ou inexistência de testes:

- Iniciar com smoke tests para validar disponibilidade do sistema.
- Construir cobertura incremental priorizando riscos.
- Documentar cenários não cobertos e criar plano de evolução.
- Mostrar valor dos primeiros testes automatizados para engajar o time.