

Melhorando o planejamento de testes da challenge -01

O que foi melhorado:

De acordo com o feedback repassado pelo instrutor QA, foram feitas as seguintes melhorias no planejamento de testes.

- ✓ Ampliação de testes para /usuarios e /carrinho.
- ✓ Colocado com mais detalhes as tabelas de matriz de risco e cobertura de testes.
- ✓ Foi feito no planejamento melhorias para as mensagens de erro, como pedido.
- ✓ Foi feito alguns cenários de bordo, conforme pedido, para trazer confiabilidade.

Prioridades das melhorias

- ✓ Para melhor entendimento, caso outro QA pegue o meu planejamento.
 - ✓ Prioridade na segurança a ser implementada.
-

Segue abaixo o novo planejamento de testes melhorado: 🔍

Cenários de Teste Planejados (Ampliados)

1. Produto

1.1 Criar produto com dados válidos (POST)

Pré-condição: Usuário autenticado como vendedor.

Ação: Enviar POST para `/produtos` com dados válidos.

Resultado esperado:

- Produto cadastrado com sucesso, HTTP 201.
- Resposta retorna ID e dados cadastrados.
- Campos obrigatórios preenchidos corretamente.

Cenários de borda:

- Preço = 0
- Nome muito longo
- Descrição vazia ou muito longa
- Tags ou categorias extras

Mensagens de erro reforçadas:

- “Preço deve ser maior que zero”
 - “Nome é obrigatório”
 - “Formato de campo inválido”
-

1.2 Criar produto com dados inválidos (POST)

Pré-condição: Usuário autenticado como vendedor.

Ação: Enviar POST com dados inválidos (preço negativo, nome vazio, campos ausentes).

Resultado esperado:

- Produto **não cadastrado**, HTTP 400.
- Mensagem de erro clara e específica para cada falha.

Cenários de borda:

- Preço como texto
 - Nome com caracteres especiais
 - Campos extras inesperados
-

1.3 Atualizar produto inexistente (PUT)

Pré-condição: Produto não existe.

Ação: Enviar PUT para `/produtos/{id}` .

Resultado esperado:

- Novo produto criado, HTTP 201.
- Resposta retorna ID e dados enviados.

Cenários de borda:

- ID inválido (texto, caracteres especiais)
 - Dados parcialmente válidos
-

2. Usuário

2.1 Criar usuário com dados válidos (POST)

- Pré-condição: Nenhum usuário com mesmo e-mail.
- Ação: POST `/usuarios` com dados válidos.
- Resultado esperado: Usuário cadastrado, HTTP 201, ID retornado.

2.2 Criar usuário com dados inválidos (POST)

- Dados duplicados ou obrigatórios faltando.
 - Resultado esperado: HTTP 400 com mensagem de erro específica.
 - Cenários de borda: e-mail inválido, senha curta, caracteres especiais no nome.
-

3. Carrinho

3.1 Adicionar produto ao carrinho (POST)

- Pré-condição: Usuário autenticado, produto existente.
- Ação: POST `/carrinhos/{id}/produtos` .
- Resultado esperado: Produto adicionado, HTTP 201, detalhes do carrinho atualizados.

3.2 Adicionar produto inexistente

- Resultado esperado: HTTP 404 com mensagem de erro clara: “Produto não encontrado”.

3.3 Remover produto do carrinho (DELETE)

- Pré-condição: Produto existe no carrinho.
- Resultado esperado: Produto removido, HTTP 200, carrinho atualizado.

Cenários de borda:

- Carrinho vazio ao tentar remover
 - Adicionar produto duplicado
 - IDs inválidos de carrinho ou produto
-

Priorização da Execução dos Cenários

Cenário	Prioridade	Justificativa
Criar produto com dados válidos (POST)	Alta	Fluxo principal da API de produtos; falha aqui compromete todo o cadastro.
Criar produto com dados inválidos (POST)	Alta	Garante validação de dados; falha pode permitir produtos inválidos.
Atualizar produto inexistente (PUT)	Média	Exceção importante, mas não impede funcionamento normal.

Criar usuário com dados válidos (POST)	Alta	Fluxo crítico de registro de usuário; impede cadastro duplicado.
Criar usuário com dados inválidos (POST)	Alta	Garantia de integridade de dados do usuário.
Adicionar produto ao carrinho (POST)	Alta	Fluxo principal de uso do carrinho; falha impacta experiência do usuário.
Adicionar produto inexistente ao carrinho	Média	Exceção; importante validar mensagens de erro.
Remover produto do carrinho (DELETE)	Média	Garantir consistência do carrinho; falha não bloqueia outros fluxos.

Matriz de Risco

Cenário	Impacto	Probabilidade	Classificação
Criar produto com dados válidos	Alto	Alta	Crítico
Criar produto com dados inválidos	Alto	Alta	Crítico
Atualizar produto inexistente	Médio	Média	Alto
Criar usuário com dados válidos	Alto	Alta	Crítico
Criar usuário com dados inválidos	Alto	Alta	Crítico
Adicionar produto ao carrinho	Alto	Alta	Crítico
Adicionar produto inexistente ao carrinho	Médio	Média	Alto

Remover produto do carrinho	Médio	Média	Alto
-----------------------------	-------	-------	------

Cobertura de Testes

Cenário	Endpoint	Cobertura API
Criar produto com dados válidos	/produtos	Cadastro completo de produto; valida campos obrigatórios.
Criar produto com dados inválidos	/produtos	Validação de regras e campos obrigatórios; automatização via pre-request para dados aleatórios.
Atualizar produto inexistente	/produtos/{id}	Criação de produto quando ID não existe; IDs gerados dinamicamente.
Criar usuário com dados válidos	/usuarios	Cadastro de usuário; valida campos obrigatórios e duplicidade.
Criar usuário com dados inválidos	/usuarios	Validação de campos e regras de negócio; mensagens de erro específicas.
Adicionar produto ao carrinho	/carrinhos/{id}/produtos	Adição de produtos; valida carrinho e quantidade de itens.
Adicionar produto inexistente ao carrinho	/carrinhos/{id}/produtos	Validação de erro quando produto não existe; mensagem clara.
Remover produto do carrinho	/carrinhos/{id}/produtos/{idProduto}	Remoção de produtos; valida atualização do carrinho.