



# Cenários de Testes Planejados

## 8. Cenários de Testes Planejados

Os testes estão agrupados por funcionalidades (User Stories). Cada caso inclui cenário, ação esperada e validação dos requisitos.

### 1 US 001 - Usuários

ID	Cenário	Entrada / Ação	Resultado Esperado	Resultado Obtido
CT1	Criar usuário válido	Nome, email válido, senha entre 5 e 10 caracteres, administrador = "true"	Cadastro realizado com sucesso (201 Created).	Cadastro realizado com sucesso (201 Created).
CT2	Criar usuário com e-mail já cadastrado	E-mail já existente	Retorno 400 com mensagem de erro "Este email já está sendo usado"	Retorno 400 com mensagem de erro "Este email já está sendo usado"
CT3	Criar usuário com e-mail de domínio <a href="#">gmail.com</a>	Ex: <a href="mailto:testebatman@gmail.com">testebatman@gmail.com</a>	Retorno 400, com mensagem de erro.	Cadastro realizado com sucesso (201 Created).
CT4	<div></div> <div>Criar usuário com e-mail de domínio  Microsoft Outlook (formerly Hotmail): Free email and calendar   Microsoft 365</div>	Ex: <a href="mailto:testebatman@hotmail.com">testebatman@hotmail.com</a>	Mesmo comportamento do CT3	Cadastro realizado com sucesso (201 Created).
CT5	Criar usuário com e-mail em formato inválido	Ex: <a href="#">batmail.com</a> ou batman@@mail	Retorno 400 com mensagem de erro "Formato de e-mail inválido"	Retorno 400 com mensagem de erro "email deve ser um email válido"
CT6	Criar usuário com senha menor que 5 caracteres	Ex: "123"	Retorno 400 com mensagem erro	Cadastro realizado com sucesso (201 Created).
CT7	Criar usuário com senha maior que 10 caracteres	Ex: "12345678901"	Retorno 400 com mensagem de erro	Cadastro realizado com sucesso (201 Created).

CT8	Atualizar usuário existente com novos dados válidos	PUT + ID válido	Retorno 200 OK	Retorno 200 com mensagem "Registro alterado com sucesso"
CT9	Atualizar usuário com ID inexistente	PUT com ID inexistente	Novo usuário criado (201 Created)	Cadastro realizado com sucesso (201 Created).
CT10	Atualizar usuário com e-mail já existente (mesmo de outro usuário)	PUT com e-mail duplicado	Retorno 400 com mensagem de erro "Este email já está sendo usado"	Retorno 400 com mensagem de erro "Este email já está sendo usado"
CT11	Listar todos os usuários	GET /usuarios	Retorno 200 com lista de usuários existentes	Retorno 200 com lista de usuários existentes
CT12	Listar usuário por ID válido	GET /usuarios/:id	Retorno 200 com os dados do usuário	Retorno 200 com os dados do usuário
CT13	Listar usuário por ID inexistente	GET /usuarios/:id	Retorno 400 com mensagem de erro "Usuário não encontrado"	Retorno 400 com mensagem de erro "Usuário não encontrado"
CT14	Deletar usuário existente	DELETE /usuarios/:id	Retorno 200 OK e mensagem de sucesso	Retorno 400 com mensagem: "Não é permitido excluir usuário com carrinho cadastrado"
CT15	Deletar usuário com ID inválido ou inexistente	DELETE /usuarios/:id	Retorno 400 com mensagem "Usuário não encontrado"	Retorno 200 com mensagem: "Nenhum registro excluído"
CT16	Tentar acessar rota de usuário inexistente com autenticação	Token válido, ID inválido	Retorno 401, com mensagem de erro adequada	Retorno 400 com mensagem de erro "Usuário não encontrado"
CT17	Verificar se os campos obrigatórios (nome, email, senha, admin) são validados corretamente	POST sem nome ou sem email ou sem senha	Retorno 400 com mensagens específicas por campo	Retorno 400 com mensagens específicas por campo
CT18	Criar usuário com admin "false"	Testar o campo "administrador"	Verificar se é aceito e retorna os dados conforme esperado	Cadastro realizado com sucesso (201 Created).
CT19	Validar campos extras ou desnecessários no payload	POST com campo adicional	API retorna erro 400	API retorna erro 400
CT20	Enviar payload vazio	POST com body vazio	Retorno 400 com mensagens de erro	Retorno 400 com mensagens de erro

## 2 US 002 - Login

ID	Cenário	Entrada / Ação	Resultado Esperado	Resultado Obtido
CT21	Login com credenciais válidas	E-mail e senha corretos	Retorno 200 OK, token Bearer gerado na resposta, duração de 10 minutos	Retorno 200 OK, token Bearer gerado na resposta, duração de 10 minutos
CT22	Login com e-mail não cadastrado	E-mail inexistente, senha qualquer	Retorno 401 Unauthorized	Retorno 401 Unauthorized
CT23	Login com senha incorreta	E-mail cadastrado, senha errada	Retorno 401 Unauthorized	Retorno 401 Unauthorized
CT24	Login com e-mail em formato inválido	Ex: " <a href="http://batman.com">http://batman.com</a> "	Retorno 400 Bad Request ou 401 Unauthorized conforme especificação	Retorno 400 com mensagem "email deve ser um email válido"
CT25	Login com senha fora dos limites estabelecidos (curta ou longa)	Senha com menos de 5 ou mais de 10 caracteres	Retorno 400	Retorno 401 Unauthorized
CT26	Login com payload incompleto	Apenas e-mail ou apenas senha	Retorno 400 com mensagem de campos obrigatórios ausentes	Retorno 400 com mensagem "password é obrigatório"
CT27	Login com payload vazio	{}	Retorno 400 com mensagem de erro	Retorno 400 com mensagem <div> 1 "email": "email é obrigatório", 2 "password": "password é obrigatório" </div>
CT28	Verificar estrutura do token retornado	Após login com sucesso	Token JWT e expiração correta de 10 minutos	Mesmo resultado do CT21
CT29	Verificar se token expira após 10 minutos	Realizar login, aguardar 11 minutos, tentar acessar rota protegida com token	Retorno 401 Unauthorized	Retorno 401 Unauthorized "Token de acesso ausente, inválido, expirado ou usuário do token não existe mais"
CT30	Verificar se mensagens de erro não expõem dados sensíveis	Ex: senha incorreta	Mensagem genérica: "Usuário e/ou senha inválidos", sem especificar se o usuário existe	Mesmo resultado do CT22

### 3 US 003 - Produtos [🔗](#)

ID	Cenário	Entrada / Ação	Resultado Esperado	Resultado Obtido
CT31	Cadastrar produto com usuário autenticado	Token válido, produto com nome único, descrição, preço, quantidade	Retorno 201 Created com dados do produto	Retorno 201 Created com dados do produto
CT032	Cadastrar produto com nome já existente	Token válido, nome duplicado	Retorno 400 com mensagem "Produto já cadastrado" ou equivalente	Retorno 400 com mensagem "Produto já cadastrado" ou equivalente
CT033	Cadastrar produto sem autenticação	Sem token	Retorno 401 Unauthorized	Retorno 401 Unauthorized
CT034	Cadastrar produto com token inválido ou expirado	Token inválido	Retorno 401 Unauthorized	Retorno 401 Unauthorized
CT035	Cadastrar produto com campos obrigatórios ausentes	Sem nome ou sem preço, por exemplo	Retorno 400 Bad Request com mensagem de erro de validação	Retorno 400 Bad Request com mensagem de erro de validação
CT036	Atualizar produto com ID inexistente	PUT com ID inexistente, nome único, token válido	Produto deve ser criado – retorno 201 Created ou comportamento documentado	Retorno 201 Created "Cadastro realizado com sucesso"
CT037	Atualizar produto com nome já utilizado	PUT com nome que já existe em outro produto	Retorno 400 Bad Request	
CT038	Atualizar produto com token inválido	PUT com token expirado	Retorno 401 Unauthorized	
CT039	Excluir produto existente e fora de carrinho	DELETE com ID válido, token válido	Retorno 204 No Content	
CT040	Tentar excluir produto que está em um carrinho ativo	DELETE com produto associado a carrinho	Retorno 400 Bad Request	
CT041	Tentar excluir produto sem autenticação	DELETE sem token	Retorno 401 Unauthorized	
CT042	Listar todos os produtos	GET sem token	Retorno 401 Unauthorized	
CT043	Buscar produto por ID válido	GET com ID de produto existente	Retorno 200 OK com detalhes do produto	
CT044	Buscar produto por ID inválido	GET com ID inexistente	Retorno 404 Not Found	

### 4 API Carrinho [🔗](#)

ID	Cenário	Entrada / Ação	Resultado Esperado
----	---------	----------------	--------------------

ID	Cenário	Entrada / Ação	Resultado Esperado
CT045	Criar carrinho com um ou mais produtos válidos	Token válido, produtos existentes e com estoque	Retorno 201 Created com ID do carrinho
CT046	Criar carrinho sem autenticação	Sem token	Retorno 401 Unauthorized
CT047	Criar carrinho com produto inexistente	Produto com ID inválido	Retorno 400 Bad Request com mensagem de erro
CT048	Criar carrinho com quantidade maior que o estoque	Produto com estoque menor do que o solicitado	Retorno 400 Bad Request com erro de estoque
CT049	Listar carrinhos existentes	GET /carrinhos	Retorno 200 OK com lista de carrinhos
CT050	Buscar carrinho por ID válido	GET /carrinhos/:id	Retorno 200 OK com detalhes do carrinho
CT051	Buscar carrinho por ID inexistente	GET /carrinhos/:id com ID inválido	Retorno 404 Not Found
CT052	Excluir carrinho existente	DELETE /carrinhos/:id com token válido	Retorno 204 No Content
CT053	Excluir carrinho sem autenticação	DELETE /carrinhos/:id sem token	Retorno 401 Unauthorized
CT054	Excluir carrinho inexistente	DELETE /carrinhos/:id com ID inválido	Retorno 404 Not Found
CT055	Excluir carrinho e retornar produtos ao estoque	DELETE /carrinhos/cancelar-compra com parâmetro <code>idCarrinho</code> , com token válido	Retorno 200 OK e produtos retornam ao estoque
CT056	Verificar se produtos voltam ao estoque após cancelamento de carrinho	Criar carrinho, cancelar, verificar estoque dos produtos	Estoque do(s) produto(s) deve ser restaurado corretamente
CT057	Cancelar carrinho inexistente	DELETE /carrinhos/cancelar-compra com ID inválido	Retorno 404 Not Found

CT058	Criar novo carrinho quando o usuário já possui um carrinho ativo	POST novo carrinho com token de usuário que já tem carrinho	Retorno 400 Bad Request com mensagem "Usuário já possui carrinho"
-------	--	---	---