



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
Usuários							
1	Criar usuário com sucesso	1. Enviar uma requisição POST para /usuarios com os seguintes dados: { "nome": "João Silva", "email": "joao.silva@email.com", "password": "senha123", "administrador": "true" } 2. Verificar se o status retornado é 201 Created. 3. Validar se a resposta contém o ID do usuário.	{ "nome": "João Silva", "email": "joao.silva@email.com", "password": "senha123", "administrador": "true" }	201 Created	201 Created	APROVADO	
2	Criar usuário sem nome	1. Enviar uma requisição POST para /usuarios sem o campo "nome". { "email": "usuario@email.com", "password": "senha123", "administrador": "true" } 2. Verificar se o status retornado é 400 Bad Request.	{ "email": "usuario@email.com", "password": "senha123", "administrador": "true" }	400 Bad Request	400 Bad Request	APROVADO	
3	Criar usuário com e-mail já existente	1. Enviar uma requisição POST para /usuarios com um e-mail já cadastrado. { "nome": "Joao Silva Novo", "email": "joao.silva@email.com", "password": "senha123", "administrador": "true" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Joao Silva Novo", "email": "joao.silva@email.com", "password": "senha123", "administrador": "true" }	400 Bad Request	400 Bad Request	APROVADO	
4	Consultar lista de usuários	1. Enviar uma requisição GET para /usuarios. 2. Verificar se o status retornado é 200 OK e se a lista de usuários é exibida	GET para /usuarios	200 Ok	200 Ok	APROVADO	
5	Consultar usuário por ID válido	1. Enviar uma requisição GET para /usuarios/{idUserio} com um ID válido. 2. Verificar se o status retornado é 200 OK e se o usuário é exibido corretamente	GET para /usuarios/hqcepeXhTmNXyAOB	200 Ok	200 Ok	APROVADO	
6	Consultar usuário por ID inválido	1. Enviar uma requisição GET para /usuarios/{idUserio} com um ID inexistente. 2. Verificar se o status retornado é 400 Bad Request.	GET para /usuarios/{idUsuariInvalido}	400 Bad Request	400 Bad Request	APROVADO	
7	Atualizar usuário com sucesso	1. Enviar uma requisição PUT para /usuarios/{idUserio} com os seguintes dados: { "nome": "João Pedro Silva", "email": "joao.pedro@email.com", "password": "novaSenha123", "administrador": "true" } 2. Verificar se o status retornado é 200 OK e se os dados do usuário são atualizados	{ "nome": "João Pedro Silva", "email": "joao.pedro@email.com", "password": "novaSenha123", "administrador": "true" }	200 Ok	200 Ok	APROVADO	



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
8	Atualizar usuário com ID inválido	1. Enviar uma requisição PUT para /usuarios/{idUserio} com os seguintes dados: { "nome": "João Inexistente", "email": "joao.inexistente@email.com", "password": "novaSenha123", "administrador": "true" } 2. Verificar se o status retornado é 400 Bad Request e a mensagem: Usuário não encontrado é exibida	{ "nome": "João Inexistente", "email": "joao.inexistente@email.com", "password": "novaSenha123", "administrador": "true" }	400 Bad Request	200 Ok	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-161
9	Excluir usuário com sucesso	1. Enviar uma requisição DELETE para /usuarios/{idUserio}. 2. Verificar se o status retornado é 200 OK e a mensagem: Registro excluído com sucesso é exibida	DELETE para /usuarios/{idUserioValido}	200 Ok	200 Ok	APROVADO	
10	Excluir usuário com ID inexistente	1. Enviar uma requisição DELETE para /usuarios/{idInexistente}. 2. Verificar se o status retornado é 400 Bad Request.	DELETE para /usuarios/{idUserioInvalido}	400 Bad Request	200 Ok	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-162
11	Criar usuário com senha muito curta	1. Enviar uma requisição POST para /usuarios com senha menor que 6 caracteres. { "nome": "Carlos Lima", "email": "carloslima@email.com", "password": "123", "administrador": "true" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Carlos Lima", "email": "carloslima@email.com", "password": "123", "administrador": "true" }	400 Bad Request	200 Ok	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-163
12	Criar usuário sem e-mail	1. Enviar uma requisição POST para /usuarios sem o campo "email". { "nome": "Ana Clara", "password": "123456", "administrador": "true" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Ana Clara", "password": "123456", "administrador": "true" }	400 Bad Request	400 Bad Request	APROVADO	
13	Criar usuário com e-mail inválido	1. Enviar uma requisição POST para /usuarios com um e-mail inválido. { "nome": "Carlos Lima", "email": "carlos-email.com", "password": "senha123", "administrador": "true" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Carlos Lima", "email": "carlos-email.com", "password": "senha123", "administrador": "true" }	400 Bad Request	400 Bad Request	APROVADO	
14	Criar usuário sem senha	1. Enviar uma requisição POST para /usuarios sem o campo "password". { "nome": "Ana Clara", "email": "ana@email.com", "administrador": "false" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Ana Clara", "email": "ana@email.com", "administrador": "false" }	400 Bad Request	400 Bad Request	APROVADO	
15	Atualizar usuário sem passar nenhum dado novo	1. Enviar uma requisição PUT para /usuarios/{idUserio} sem modificar nenhum campo. { } 2. Verificar se o status retornado é 400 Bad Request.	{ }	400 Bad Request	400 Bad Request	APROVADO	



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
Login							
16	Login com sucesso	1. Enviar uma requisição POST para /login com as credenciais corretas. { "email": "joao.silva@email.com", "password": "senha123" } 2. Verificar se o status retornado é 200 OK. 3. Validar se a resposta contém um token de autenticação.	{ "email": "joao.silva@email.com", "password": "senha123" }	200 Ok	200 Ok	APROVADO	
17	Login com senha errada	1. Enviar uma requisição POST para /login com e-mail válido e senha incorreta. { "email": "joao.silva@email.com", "password": "senhaErrada" } 2. Verificar se o status retornado é 401 Unauthorized.	{ "email": "joao.silva@email.com", "password": "senhaErrada" }	401 Unauthorized	401 Unauthorized	APROVADO	
18	Login com e-mail inexistente	1. Enviar uma requisição POST para /login com um e-mail não cadastrado. { "email": "emailinexistente@email.com", "password": "senha123" } 2. Verificar se o status retornado é 401 Unauthorized.	{ "email": "emailinexistente@email.com", "password": "senha123" }	401 Unauthorized	401 Unauthorized	APROVADO	
19	Login sem senha	1. Enviar uma requisição POST para /login sem o campo "password". { "email": "joao.silva@email.com" } 2. Verificar se o status retornado é 400 Bad Request.	{ "email": "joao.silva@email.com" }	400 Bad Request	400 Bad Request	APROVADO	
20	Login sem e-mail	1. Enviar uma requisição POST para /login sem o campo "email". { "password": "senha123" } 2. Verificar se o status retornado é 400 Bad Request.	{ "password": "senha123" }	400 Bad Request	400 Bad Request	APROVADO	
21	Login com e-mail em caixa alta	Criar um usuário via POST em /usuarios com um e-mail em letras minúsculas. Enviar uma requisição POST para /login usando o mesmo e-mail, mas em letras maiúsculas. { "email": "USUARIO@EMAIL.COM", "password": "senha123" } Verificar se a API permite o login corretamente	{ "email": "USUARIO@EMAIL.COM", "password": "senha123" }	401 Unauthorized	401 Unauthorized	APROVADO	



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
22	Login com espaços extras no e-mail	<p>Criar um usuário via POST em /usuarios com um e-mail válido.</p> <p>Enviar uma requisição POST para /login usando o mesmo e-mail, mas adicionando espaços extras antes e/ou depois.</p> <pre>{ "email": " usuario@email.com ", "password": "senha123" }</pre> <p>Verificar se a API permite o login corretamente e ignora os espaços extras.</p>	<pre>{ "email": " usuario@email.com ", "password": "senha123" }</pre>	400 Bad Request	400 Bad Request	APROVADO	
23	Login com e-mail contendo caracteres especiais inesperados	<p>Criar um usuário via POST em /usuarios com um e-mail válido.</p> <p>Enviar uma requisição POST para /login usando o mesmo e-mail, mas adicionando caracteres especiais inesperados (como tabulação ou quebras de linha).</p> <pre>{ "email": "usuario@email.com\t", "password": "senha123" }</pre> <p>Verificar se a API trata corretamente esses caracteres ou retorna erro.</p>	<pre>{ "email": "usuario@email.com\t", "password": "senha123" }</pre>	400 Bad Request	400 Bad Request	APROVADO	
24	Logout sem token	<p>Criar um usuário via POST em /usuarios com um e-mail em letras minúsculas.</p> <p>Enviar uma requisição POST para /login usando o mesmo e-mail, mas deixando apenas o domínio em caixa alta.</p> <pre>{ "email": "usuario@EMAIL.COM", "password": "senha123" }</pre> <p>Verificar se a API trata o domínio de e-mail de forma case insensitive.</p>	<pre>{ "email": "usuario@EMAIL.COM", "password": "senha123" }</pre>	401 Unauthorized	401 Unauthorized	APROVADO	
25	Login com e-mail contendo múltiplos pontos seguidos	<p>Criar um usuário via POST em /usuarios com um e-mail válido.</p> <p>Enviar uma requisição POST para /login usando o mesmo e-mail, mas adicionando pontos extras na parte local antes do @.</p> <pre>{ "email": "usu..ario@email.com", "password": "senha123" }</pre> <p>Verificar se a API permite o login ou rejeita e-mails com formatação incorreta.</p>	<pre>{ "email": "usu..ario@email.com", "password": "senha123" }</pre>	400 Bad Request	400 Bad Request	APROVADO	
Produtos							
26	Criar produto com sucesso	<p>1. Enviar uma requisição POST para /produtos com os seguintes dados:</p> <pre>{ "nome": "Notebook Dell", "preco": 2000, "descricao": "Notebook Dell i5", "quantidade": 10 }</pre> <p>2. Verificar se o status retornado é 201 Created.</p> <p>3. Validar se o corpo da resposta contém o ID do produto e os mesmos dados enviados.</p>	<pre>{ "nome": "Notebook Dell", "preco": 2000, "descricao": "Notebook Dell i5", "quantidade": 10 }</pre>	201 Created	201 Created	APROVADO	



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
27	Criar produto sem nome	1. Enviar uma requisição POST para /produtos sem o campo "nome": { "preco": 2000, "descricao": "Notebook Dell i5", "quantidade": 10 } 2. Verificar se o status retornado é 400 Bad Request. 3. Validar se a resposta contém uma mensagem de erro.	{ "preco": 2000, "descricao": "Notebook Dell i5", "quantidade": 10 }	400 Bad Request	400 Bad Request	APROVADO	
28	Criar produto com nome já existente	1. Enviar um POST para /produtos com um nome de produto já cadastrado: { "nome": "Notebook Dell", "preco": 2000, "descricao": "Notebook Dell i5", "quantidade": 10 } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Notebook Dell", "preco": 2000, "descricao": "Notebook Dell i5", "quantidade": 10 }	400 Bad Request	400 Bad Request	APROVADO	
29	Criar produto com preço negativo	1. Enviar um POST para /produtos com preço negativo: { "nome": "Notebook Dell", "preco": -2000, "descricao": "Notebook Dell i5", "quantidade": 10 } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Notebook Dell", "preco": -2000, "descricao": "Notebook Dell i5", "quantidade": 10 }	400 Bad Request	400 Bad Request	APROVADO	
30	Criar produto sem preço	1. Enviar um POST sem o campo "preco": { "nome": "Mouse sem fio", "descricao": "Mouse sem fio com tecnologia Bluetooth", "quantidade": 8 } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Mouse sem fio", "descricao": "Mouse sem fio com tecnologia Bluetooth", "quantidade": 8 }	400 Bad Request	400 Bad Request	APROVADO	
31	Criar produto sem quantidade	1. Enviar um POST sem o campo "quantidade". { "nome": "Notebook Dell", "preco": 2000, "descricao": "Notebook Dell i5" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Notebook Dell", "preco": 2000, "descricao": "Notebook Dell i5" }	400 Bad Request	400 Bad Request	APROVADO	
32	Criar produto com quantidade negativa	1. Enviar um POST com quantidade negativa: { "nome": "Cadeira Gamer", "preco": 800, "descricao": "Cadeira ergonômica para jogos", "quantidade": -3 } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Cadeira Gamer", "preco": 800, "descricao": "Cadeira ergonômica para jogos", "quantidade": -3 }	400 Bad Request	400 Bad Request	APROVADO	
33	Consultar lista de produtos	1. Enviar uma requisição GET para /produtos. 2. Verificar se o status retornado é 200 OK. 3. Validar se a resposta contém uma lista de produtos.		200 OK	200 OK	APROVADO	



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
34	Consultar produto por ID válido	1. Enviar uma requisição GET para /produtos/SejMYTVuXXMIFpEz com um ID válido. 2. Verificar se o status retornado é 200 OK. 3. Validar se a resposta contém os detalhes do produto solicitado.	GET para /produtos/SejMYTVuXXMIFpEz	200 OK	200 OK	APROVADO	
35	Consultar produto por ID inexistente	1. Enviar uma requisição GET para /produtos/{id} com um ID inexistente. 2. Verificar se o status retornado é 400 Bad Request.		400 Bad Request	400 Bad Request	APROVADO	
36	Atualizar produto com sucesso	1. Enviar uma requisição PUT para /produtos/{id} com um ID válido. { "nome": "Notebook Dell Atualizado", "preco": 1900, "descricao": "Notebook Dell i5", "quantidade": 10 } 2. Verificar se o status retornado é 200 OK. 3. Validar se os detalhes do produto foram atualizados.	{ "nome": "Notebook Dell Atualizado", "preco": 1900, "descricao": "Notebook Dell i5", "quantidade": 10 }	200 OK	200 OK	APROVADO	
37	Atualizar produto com ID inexistente	1. Enviar uma requisição PUT para /produtos/{id} com um ID inexistente. { "nome": "Produto Inexistente", "preco": 100, "descricao": "Produto Inexistente", "quantidade": 10 } 2. Verificar se o status retornado é 404 Not Found.	{ "nome": "Produto Inexistente", "preco": 100, "descricao": "Produto Inexistente", "quantidade": 10 }	404 Not Found	201 Created	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-164
38	Excluir produto com sucesso	1. Enviar uma requisição DELETE para /produtos/SejMYTVuXXMIFpEz com um ID válido. 2. Verificar se o status retornado é 200 OK. 3. Validar se o produto foi removido corretamente ao tentar buscá-lo novamente.	DELETE para /produtos/SejMYTVuXXMIFpEz	200 OK	200 OK	APROVADO	
39	Excluir produto com ID inexistente	1. Enviar uma requisição DELETE para /produtos/{id} com um ID inexistente. 2. Verificar se o status retornado é 404 Not Found.		404 Not Found	200 OK	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-165
40	Criar produto com descrição vazia	1. Enviar uma requisição POST para /produtos com um corpo de requisição sem a descrição do produto. { "nome": "Novo Produto", "preco": 200.0, "descricao": "" } 2. Verificar se o status retornado é 400 Bad Request.	{ "nome": "Novo Produto", "preco": 200.0, "descricao": "" }	400 Bad Request	400 Bad Request	APROVADO	
Carrinho de Compras							



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
41	Adicionar produto ao carrinho com sucesso	1. Enviar um POST para /carrinhos com um produto válido: { "produtos": [{ "idProduto": "m3xg2CctxeQBgeXY", "quantidade": 2 }] } 2. Verificar se o status retornado é 201 Created.	{ "produtos": [{ "idProduto": "m3xg2CctxeQBgeXY", "quantidade": 2 }] }	201 Created	201 Created	APROVADO	
42	Adicionar produto inexistente ao carrinho	1. Enviar um POST para /carrinhos com um ID de produto inexistente: { "produtos": [{ "idProduto": "000000", "quantidade": 2 }] } 2. Verificar se o status retornado é 400 Bad Request.	{ "produtos": [{ "idProduto": "000000", "quantidade": 2 }] }	400 Bad Request	400 Bad Request	APROVADO	
43	Adicionar produto com quantidade inválida	1. Enviar uma requisição POST para /carrinho com um produto e quantidade inválida (por exemplo, quantidade negativa ou zero). { "produtos": [{ "idProduto": "m3xg2CctxeQBgeXY", "quantidade": -1 }] } 2. Verificar se o status retornado é 400 Bad Request.	{ "produtos": [{ "idProduto": "m3xg2CctxeQBgeXY", "quantidade": -1 }] }	400 Bad Request	400 Bad Request	APROVADO	
44	Consultar carrinho de compras com itens	1. Adicionar um ou mais produtos ao carrinho via POST. 2. Enviar uma requisição GET para /carrinho. 3. Verificar se o status retornado é 200 OK. 4. Validar se a resposta contém os produtos adicionados no carrinho.		200 OK, lista dos itens no carrinho.	200 OK, lista dos itens no carrinho.	APROVADO	
45	Consultar carrinho vazio	1. Enviar uma requisição GET para /carrinho sem ter adicionado produtos. 2. Verificar se o status retornado é 200 OK. 3. Validar se a resposta contém uma lista vazia de produtos.		200 Ok	200 Ok	APROVADO	
46	Adicionar o mesmo produto ao carrinho mais de uma vez	1. Criar um usuário via POST em /usuarios. 2. Realizar login via POST em /login e obter o token de autenticação. 3. Consultar a lista de produtos via GET em /produtos e selecionar um ID válido. 4. Enviar uma requisição POST para /carrinhos, adicionando o mesmo produto duas vezes.	{ "produtos": [{ "idProduto": "m3xg2CctxeQBgeXY", "quantidade": 1 }] }	200 Ok	400 Bad Request	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-166



Id do Teste	Descrição do teste	Etapas	Dados do teste (Corpo da solicitação)	Resultado esperado	Resultado real	Status	Link para o relatório de bug
47	Adicionar produto ao carrinho sem informar a quantidade	1. Criar um usuário via POST em /usuarios. 2. Realizar login via POST em /login e obter o token de autenticação. 3. Consultar a lista de produtos via GET em /produtos e selecionar um ID válido. 4. Enviar uma requisição POST para /carrinhos, mas sem passar o campo "quantidade". { "produtos": [{ "idProduto": "produto1" }] } 5. Verificar se o status retornado é 400 Bad Request e validar a mensagem de erro.	{ "produtos": [{ "idProduto": "produto1" }] }	400 Bad Request	400 Bad Request	APROVADO	
48	Tentar adicionar um produto ao carrinho com estoque insuficiente	1. Criar um usuário via POST em /usuarios. 2. Realizar login via POST em /login e obter o token de autenticação. 3. Consultar a lista de produtos via GET em /produtos e selecionar um ID válido. 4. Verificar o estoque do produto e tentar adicioná-lo ao carrinho com uma quantidade superior ao estoque disponível via POST em /carrinhos. { "produtos": [{ "idProduto": "produto1", "quantidade": 9999 }] } 5. Verificar se a API retorna um erro indicando que não há estoque suficiente.	{ "produtos": [{ "idProduto": "produto1", "quantidade": 9999 }] }	400 Bad Request	400 Bad Request	APROVADO	
49	Tentar consultar carrinho de compras sem estar autenticado	1. Criar um usuário via POST em /usuarios. 2. Realizar login via POST em /login e obter o token de autenticação. 3. Adicionar produtos ao carrinho via POST em /carrinhos. 4. Tentar consultar o carrinho enviando uma requisição GET para /carrinhos sem enviar o token de autenticação. 5. Verificar se a API retorna um erro de autenticação.		401 Unauthorized	200 Ok	REPROVADO	https://elainemtaraujo.atlassian.net/browse/BR-167
50	Limpar carrinho de compras	1. Adicionar um ou mais produtos ao carrinho via POST. 2. Enviar uma requisição DELETE para /carrinho/cancelar-compra para remover todos os itens do carrinho. 3. Verificar se o status retornado é 200 OK. 4. Validar se o carrinho foi limpo com sucesso, ou seja, não contém mais produtos.	DELETE para /carrinho/cancelar-compra	200 OK, carrinho esvaziado.	200 OK, carrinho esvaziado.	APROVADO	