

/USUARIOS				
ID	Cenário de Teste	Resultado Esperado	Resultado Obtido	Status
US-01	Buscar usuário por ID válido	Retornar os dados do usuário	200: informações do usuário retornadas	✓ Correto
US-02	Buscar usuário por ID inexistente	Retornar erro informando que não existe	400: "Usuário não encontrado"	✓ Correto
US-03	Cadastrar usuário válido	Usuário deve ser cadastrado com sucesso	201: "Cadastro realizado com sucesso" + ID	✓ Correto
US-04	Cadastrar usuário com e-mail Gmail ou Hotmail	Sistema deveria bloquear o cadastro	201: cadastro realizado → Bug	✗ Bug
US-05	Cadastrar usuário com senha menor que 5 caracteres	Sistema deveria bloquear o cadastro	201: cadastro realizado → Bug	✗ Bug
US-06	Cadastrar usuário com senha maior que 10 caracteres	Sistema deveria bloquear o cadastro	201: cadastro realizado → Bug	✗ Bug
US-07	Cadastrar usuário duplicado	Bloquear cadastro e avisar que já existe	400: "Este email já está sendo usado"	✓ Correto
US-08	Cadastrar usuário com e-mail inválido	Retornar erro indicando formato incorreto	400: "email deve ser um email válido"	✓ Correto
US-09	Atualizar usuário existente com body correto	Usuário deve ser atualizado com sucesso	200: "Registro alterado com sucesso"	✓ Correto
US-10	Atualizar usuário inexistente	Retornar erro informando que não existe	201: cria novo usuário automaticamente → ⚠ Pode ser melhorado	⚠ Melhoria
US-11	Atualizar campo administrador com valor inválido	Retornar erro informando os valores aceitos	400: "administrador deve ser 'true' ou 'false'"	✓ Correto
US-12	Atualizar usuário sem enviar body	Sistema deve informar campos obrigatórios	400: mensagens indicando campos obrigatórios	✓ Correto
US-13	Deletar usuário existente	Usuário deve ser excluído com sucesso	200: "Registro excluído com sucesso"	✓ Correto

US-14	Deletar usuário já excluído	Informar que não há nada para excluir	200: "Nenhum registro excluído"	⚠ Mensagem pode ser mais clara
US-15	Deletar usuário inexistente	Informar que não há nada para excluir	200: "Nenhum registro excluído"	⚠ Mensagem pode ser mais clara

/LOGIN					
ID	Cenário	Entrada/Descrição	Resultado Esperado	Resultado Obtido	Status
CT-L01	Login com dados corretos	POST /login com email e senha válidos	200 OK + token Bearer	200 OK – token gerado com sucesso	✅ OK
CT-L02	Login sem body	POST /login sem email/senha no body	400 Bad Request – campos obrigatórios	400 – mensagem exibindo que email e password são obrigatórios	✅ OK
CT-L03	Login com senha incorreta	POST /login com email válido e senha inválida	401 Unauthorized	401 Unauthorized – “Email e/ou senha inválidos”	✅ OK
CT-L04	Login com email inválido	POST /login com formato inválido de email	400 Bad Request	400 – “email deve ser um email válido”	✅ OK

/PRODUTOS				
ID	Cenário de Teste	Resultado Esperado	Resultado Obtido	Status
PR-01	Cadastrar produto com autenticação expirada	Retornar erro de autenticação	Erro 401: "Token de acesso ausente, inválido, expirado ou usuário do token não existe mais"	✅ Correto
PR-02	Cadastrar produto já existente	Retornar erro informando que o produto já existe	Erro 400: "Já existe produto com esse nome" → mensagem poderia ser mais clara	⚠ Melhoria
PR-03	Cadastrar produto válido com token válido	Produto cadastrado com sucesso (201)	201: "Cadastro realizado com sucesso" + ID do produto	✅ Correto

PR-04	Buscar produto por ID válida	Retornar informações do produto (200)	200 com dados do produto	✓ Correto
PR-05	Buscar produto com ID inexistente	Retornar erro informando que produto não existe	Erro 400 : "Produto não encontrado"	✓ Correto
PR-06	Atualizar produto válido	Produto atualizado com sucesso (200)	200 : "Registro alterado com sucesso"	✓ Correto
PR-07	Atualizar produto com ID inválido	Retornar erro informando que ID não existe	400 : "Já existe produto com esse nome" → provavelmente um bug (mensagem incorreta)	✗ Bug
PR-08	Atualizar produto com token de usuário não-admin	Retornar erro de permissão	403 : "Rota exclusiva para administradores"	✓ Correto
PR-09	Cadastrar produto com token de usuário não-admin	Retornar erro de permissão	403 : "Rota exclusiva para administradores"	✓ Correto
PR-10	Deletar produto válido	Produto excluído com sucesso (200)	200 : "Registro excluído com sucesso"	✓ Correto
PR-11	Deletar produto com ID inexistente	Retornar erro ou aviso claro que não foi excluído	200 : "Nenhum registro excluído com sucesso" → mensagem confusa	⚠ Melhoria
PR-12	Buscar todos os produtos cadastrados	Retornar lista de produtos (200)	200 : lista de todos os produtos cadastrados	✓ Correto

/CARRINHOS				
ID	Cenário de Teste	Resultado Esperado	Resultado Obtido	Status
CA-01	Buscar todos os carrinhos (GET)	Retornar lista de carrinhos (200)	200 : lista dos carrinhos existentes	✓ Correto
CA-02	Buscar carrinho por ID válida (GET)	Retornar lista de produtos do carrinho (200)	200 : lista dos produtos dentro do carrinho	✓ Correto
CA-03	Buscar carrinho por ID inexistente (GET)	Retornar erro informando que não existe	400 : "Carrinho não encontrado"	✓ Correto

CA-04	Cadastrar carrinho válido (POST)	Criar carrinho e retornar sucesso (201)	201: "Cadastro realizado com sucesso" + ID do carrinho	✓ Correto
CA-05	Cadastrar carrinho repetido (POST)	Retornar erro indicando que já existe carrinho	400: "Não é permitido ter mais de 1 carrinho"	✓ Correto
CA-06	Adicionar produto além da quantidade disponível (POST)	Retornar erro informando falta de estoque	400: "Produto não possui quantidade suficiente"	✓ Correto
CA-07	Concluir compra (DELETE)	Excluir carrinho e retornar sucesso (200)	200: "Registro excluído com sucesso"	✓ Correto
CA-08	Concluir compra novamente no mesmo usuário (DELETE)	Retornar erro ou aviso de que não há carrinho	200: "Não foi encontrado carrinho para esse usuário"	⚠ Melhoria (mensagem poderia ser mais clara)
CA-09	Cancelar compra (DELETE)	Cancelar carrinho e reabastecer estoque (200)	200: "Registro excluído com sucesso. Estoque dos produtos reabastecido"	✓ Correto
CA-10	Cancelar compra novamente no mesmo usuário (DELETE)	Retornar erro ou aviso de que não há carrinho	200: "Não foi encontrado carrinho para esse usuário"	⚠ Melhoria (mensagem poderia ser mais clara)
CA-11	Cancelar compra com token inválido/expirado (DELETE)	Retornar erro de autenticação (401)	401: "Token de acesso ausente, inválido, expirado ou usuário do token não existe mais"	✓ Correto

Relatórios De Bugs						
ID	Título	Descrição	Passos para Reprodução	Resultado Obtido	Resultado Esperado	Prioridade
BUG-01	Cadastro permitido com Gmail/Hotmail	O sistema permite cadastrar usuários com provedores de email proibidos.	1. POST /usuarios com email teste@gmail.com 2. Verificar resposta	201 Created – cadastro realizado	Bloquear cadastro com emails de provedores Gmail/Hotmail	Alta
BUG-02	Cadastro permitido com senha < 5 caracteres	O sistema permite cadastrar usuários com senha curta.	1. POST /usuarios com senha 123 2. Verificar resposta	201 Created – cadastro realizado	Bloquear cadastro com senha menor que 5	Alta
BUG-03	Cadastro permitido com senha > 10 caracteres	O sistema permite cadastrar usuários com senha longa.	1. POST /usuarios com senha 12345678901 2. Verificar resposta	201 Created – cadastro realizado	Bloquear cadastro com senha maior que 10	Alta
BUG-04	Atualização de produto com ID inválido retorna mensagem incorreta	Mensagem de erro não corresponde ao problema.	1. PUT /produtos/{id_invalido} 2. Verificar resposta	400 – "Já existe produto com esse nome"	Retornar mensagem clara: "Produto não encontrado" ou "ID inválido"	Média

Relatório De Melhorias						
ID	Título	Descrição	Passos para Reprodução	Resultado Obtido	Resultado Esperado	Prioridade
MEL-01	Atualização de usuário inexistente cria novo usuário	O sistema cria usuário automaticamente ao tentar atualizar um inexistente.	1. PUT /usuarios/{id_invalido} 2. Verificar resposta	201 Created – novo usuário criado	Retornar erro informando que o usuário não existe	Média
MEL-02	Mensagem ao deletar usuário já excluído	Mensagem pode causar confusão.	1. DELETE /usuarios/{id_valido} (2x) 2. Verificar resposta	200 – "Nenhum registro excluído"	Mensagem mais clara, ex.: "Usuário já excluído"	Baixa
MEL-03	Mensagem ao deletar usuário inexistente	Mensagem pode causar confusão.	1. DELETE /usuarios/{id_invalido} 2. Verificar resposta	200 – "Nenhum registro excluído"	Mensagem mais clara, ex.: "Usuário não encontrado"	Baixa
MEL-04	Mensagem ao cadastrar produto duplicado	Mensagem poderia ser mais específica.	1. POST /produtos com nome já existente 2. Verificar resposta	400 – "Já existe produto com esse nome"	Mensagem clara: "Produto já cadastrado"	Média
MEL-05	Mensagem ao deletar produto inexistente	Mensagem pouco clara.	1. DELETE /produtos/{id_invalido} 2. Verificar resposta	200 – "Nenhum registro excluído com sucesso"	Mensagem clara: "Produto não encontrado"	Baixa
MEL-06	Mensagem ao concluir compra sem carrinho	Mensagem pode ser melhorada.	1. DELETE /carrinhos concluir compra sem carrinho 2. Verificar resposta	200 – "Não foi encontrado carrinho para esse usuário"	Mensagem clara: "Usuário não possui carrinho ativo"	Média
MEL-07	Mensagem ao cancelar compra sem carrinho	Mensagem pode ser melhorada.	1. DELETE /carrinhos cancelar compra sem carrinho 2. Verificar resposta	200 – "Não foi encontrado carrinho para esse usuário"	Mensagem clara: "Usuário não possui carrinho ativo"	Média

Bugs

ID da Issue: ISS-001

Título: Cadastro permitido com Gmail/Hotmail

Descrição: O sistema permite cadastrar usuários com provedores de email proibidos (Gmail e Hotmail).

Passos para Reproduzir:

1. Fazer uma requisição POST para /usuarios.
2. Informar um email do tipo teste@gmail.com.
3. Verificar resposta da API.

Resultado Obtido: 201 Created – usuário cadastrado com sucesso.

Resultado Esperado: A API deveria bloquear o cadastro e retornar erro 400.

Gravidade/Prioridade: Alta.

ID da Issue: ISS-002

Título: Cadastro permitido com senha menor que 5 caracteres

Descrição: O sistema permite cadastrar usuários com senha menor que 5 dígitos, contrariando as regras definidas.

Passos para Reproduzir:

1. Fazer uma requisição POST para /usuarios.
2. Informar senha com 3 caracteres, ex.: "123".
3. Verificar resposta da API.

Resultado Obtido: 201 Created – usuário cadastrado com sucesso.

Resultado Esperado: A API deveria retornar erro 400, informando que a senha deve ter no mínimo 5 caracteres.

Gravidade/Prioridade: Alta.

ID da Issue: ISS-003

Título: Cadastro permitido com senha maior que 10 caracteres

Descrição: O sistema permite cadastrar usuários com senha maior que 10 dígitos, contrariando as regras definidas.

Passos para Reproduzir:

1. Fazer uma requisição POST para /usuarios.
2. Informar senha com 11 caracteres, ex.: "12345678901".
3. Verificar resposta da API.

Resultado Obtido: 201 Created – usuário cadastrado com sucesso.

Resultado Esperado: A API deveria retornar erro 400, informando que a senha deve ter no máximo

10 caracteres.

Gravidade/Prioridade: Alta.

ID da Issue: ISS-004

Título: Atualização de produto com ID inválido retorna mensagem incorreta

Descrição: Ao atualizar um produto com ID inexistente, a API retorna mensagem incorreta de erro.

Passos para Reproduzir:

1. Fazer uma requisição PUT para /produtos/{id_invalido}.
2. Enviar body com dados de produto.
3. Verificar resposta da API.

Resultado Obtido: 400 – mensagem: "Já existe produto com esse nome".

Resultado Esperado: A API deveria retornar mensagem clara: "Produto não encontrado" ou "ID inválido".

Gravidade/Prioridade: Média.

Melhorias

ID da Melhoria: MEL-001

Título: Melhorar comportamento ao atualizar usuário inexistente

Descrição: Atualmente, ao atualizar um usuário inexistente, a API cria um novo usuário automaticamente.

Benefício esperado: Evitar criação de usuários acidentalmente e garantir mais segurança e clareza para quem usa a API.

ID da Melhoria: MEL-002

Título: Mensagem mais clara ao deletar usuário já excluído

Descrição: Hoje, quando o usuário já foi excluído, a API retorna "Nenhum registro excluído".

Benefício esperado: A mensagem poderia ser mais clara, como "Usuário já excluído", para evitar confusão.

ID da Melhoria: MEL-003

Título: Mensagem mais clara ao deletar usuário inexistente

Descrição: Quando é feita uma requisição para deletar um usuário inexistente, a API retorna "Nenhum registro excluído".

Benefício esperado: Melhorar a clareza da resposta para "Usuário não encontrado".

ID da Melhoria: MEL-004

Título: Mensagem mais clara ao cadastrar produto duplicado

Descrição: Hoje a API retorna "Já existe produto com esse nome".

Benefício esperado: Melhorar clareza com mensagem como "Produto já cadastrado".

ID da Melhoria: MEL-005

Título: Mensagem mais clara ao deletar produto inexistente

Descrição: Quando um produto inexistente é deletado, a API retorna "Nenhum registro excluído com sucesso".

Benefício esperado: Melhorar clareza da mensagem para "Produto não encontrado".

ID da Melhoria: MEL-006

Título: Mensagem mais clara ao concluir compra sem carrinho ativo

Descrição: Hoje a API retorna "Não foi encontrado carrinho para esse usuário".

Benefício esperado: Melhorar clareza para "Usuário não possui carrinho ativo".

ID da Melhoria: MEL-007

Título: Mensagem mais clara ao cancelar compra sem carrinho ativo

Descrição: Hoje a API retorna "Não foi encontrado carrinho para esse usuário".

Benefício esperado: Melhorar clareza para "Usuário não possui carrinho ativo".