Tabela de Cobertura

Critério de Aceitação	Caso de Teste	Cobertura	Passo a Passo para Automação
Os vendedores (usuários) deverão possuir os campos NOME, E-MAIL, PASSWORD e ADMINISTRADOR	U01 - Criar usuário válido	O teste valida a criação de um usuário com todos os campos.	1. Montar payload JSON com todos os campos válidos 2. Enviar requisição POST para `/usuarios` 3. Validar status 201 (Created) 4. Validar corpo da resposta contém os dados criados 5. Salvar ID retornado para uso em testes futuros
Não deve ser possível criar um usuário com e-mail já utilizado	U02 - Criar usuário com e-mail duplicado	O teste valida a criação de um usuário com e-mail duplicado, impedindo a ação.	1. Criar usuário previamente com e-mail X 2. Tentar criar outro usuário com mesmo e-mail X 3. Validar status 400 ou erro esperado 4. Validar mensagem de erro indicando duplicidade
Não deverá ser possível cadastrar usuários com e-mails de provedor gmail e hotmail	U03 - Criar usuário com domínio gmail/hotmail	O teste valida que e-mails de provedores específicos são bloqueados.	1. Criar payload com e-mail @gmail.com 2. Enviar POST `/usuarios` 3. Validar status 400 ou erro esperado 4. Validar mensagem indicando restrição de domínio 5. Repetir para @hotmail.com
Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado	U04 - Atualizar usuário inexistente (criar novo)	O teste valida a atualização de um usuário inexistente, criando um novo.	1. Tentar atualizar usuário com ID inexistente via PUT `/usuarios/:id` 2. Validar status 201 (Created) ou status definido pelo requisito

			3. Validar que o usuário foi realmente criado 4. Validar dados retornados
Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT	U02 - Criar usuário com e-mail duplicado (via PUT)	O teste também se aplica ao PUT, validando a duplicação de e-mails.	1. Criar usuário inicial com e-mail X 2. Tentar atualizar outro usuário (via PUT) para e-mail X 3. Validar status 400 ou erro esperado 4. Validar mensagem de erro
As senhas devem possuir no mínimo 5 caracteres e no máximo 10 caracteres	U05 - Validar senha fora do padrão	O teste valida as restrições de tamanho para a senha.	1. Criar payload com senha < 5 caracteres 2. Enviar POST `/usuarios` 3. Validar erro e mensagem 4. Repetir para senha > 10 caracteres

Endpoint: `/login`

Critério de Aceitação	Caso de Teste	Cobertura	Passo a Passo para Automação
Usuários não cadastrados não deverão conseguir autenticar	L02 - Autenticar usuário inexistente	O teste valida que usuários não cadastrados não podem autenticar.	1. Montar payload com e-mail/senha inválidos 2. Enviar POST `/login` 3. Validar status 401 4. Validar mensagem de erro
Usuários existentes e com a senha correta deverão ser autenticados	L01 - Autenticar usuário válido	O teste valida que usuários com dados corretos são autenticados com sucesso.	1. Criar usuário válido previamente 2. Montar payload correto 3. Enviar POST `/login` 4. Validar status 200 5. Validar que resposta contém `token` 6. Validar formato do token (Bearer)
Usuários com senha inválida não deverão conseguir autenticar	L03 - Autenticar com senha inválida	O teste valida que senhas inválidas resultam em falha de autenticação.	1. Criar usuário válido previamente 2. Montar payload com e-mail correto, senha

			errada 3. Enviar POST `/login` 4. Validar status 401 5. Validar mensagem de erro
A autenticação deverá gerar um token Bearer	L01 - Autenticar usuário válido	O teste valida a geração do token Bearer.	**Mesmos passos de L01 acima, incluindo checagem do prefixo "Bearer "**
A duração da validade do token deverá ser de 10 minutos	L01 - Validar duração do token	O teste pode verificar a duração do token.	1. Obter token via login válido 2. Realizar chamadas autenticadas dentro de 10 minutos (validar sucesso) 3. Esperar 10+ minutos e repetir chamada (validar expiração: 401 ou token inválido)

Endpoint: `/produtos`

Critério de Aceitação	Caso de Teste	Cobertura	Passo a Passo para Automação
Usuários não autenticados não devem conseguir realizar ações na rota de Produtos	P07 - Operação em produto sem autenticação	O teste valida que usuários não autenticados não podem operar com produtos.	1. Enviar POST `/produtos` sem token 2. Validar status 401 3. Validar mensagem de erro
Não deve ser possível realizar o cadastro de produtos com nomes já utilizados	P02 - Criar produto com nome duplicado	O teste valida que produtos não podem ter nomes duplicados.	1. Criar produto com nome X 2. Tentar criar outro produto com nome X 3. Validar status 400 4. Validar mensagem de erro
Não deve ser possível excluir produtos que estão dentro de carrinhos	P06 - Deletar produto que está dentro de carrinho	O teste valida a impossibilidade de deletar produtos em carrinhos.	1. Criar produto 2. Adicionar produto a um carrinho 3. Tentar DELETE `/produtos/:id` 4. Validar status 400 ou definido 5. Validar mensagem indicando bloqueio

Caso não exista produto com o ID informado na hora do UPDATE, um novo produto deverá ser criado	P03 - Atualizar produto inexistente (criar novo)	O teste cobre o uso do PUT para criar novos produtos com nome único.	1. Tentar PUT `/produtos/:idInexistente` 2. Validar status 201 ou equivalente 3. Validar que produto foi criado

---

Endpoint: `/carrinhos`

Critério de Aceitação	Caso de Teste	Cobertura	Passo a Passo para Automação
Apenas usuários autenticados poderão criar e gerenciar carrinhos	C01 - Criar carrinho autenticado	O teste valida que apenas usuários autenticados podem criar e gerenciar carrinhos.	1. Autenticar e obter token 2. Montar payload válido 3. Enviar POST `/carrinhos` com token 4. Validar status 201 e conteúdo retornado
Não deve ser possível adicionar produtos inexistentes ao carrinho	C03 - Adicionar produto inexistente	O teste valida que produtos inexistentes não podem ser adicionados ao carrinho.	1. Montar payload com ID de produto inválido 2. Enviar POST `/carrinhos` 3. Validar status 400 ou erro esperado
Não deve ser possível criar carrinhos com quantidade negativa ou zero de produtos	C02 - Criar carrinho com quantidade inválida	O teste valida que a criação de carrinhos com quantidade inválida é bloqueada.	1. Montar payload com quantidade 0 ou negativa 2. Enviar POST `/carrinhos` 3. Validar status 400 ou erro esperado
Ao deletar um carrinho, os produtos associados deverão ser liberados	C06 - Deletar carrinho existente	O teste valida que produtos em carrinhos excluídos são liberados.	1. Criar carrinho com produto X 2. DELETE `/carrinhos/:id` 3. Criar novo carrinho com mesmo produto X (validar sucesso)
Não deve ser possível editar carrinhos fechados (status finalizado)	C07 - Tentar editar carrinho finalizado	O teste valida que carrinhos finalizados não podem ser editados.	1. Criar carrinho e marcar status finalizado 2. Tentar PUT `/carrinhos/:id` 3. Validar status 400 ou erro esperado
Cada usuário deve visualizar apenas seu próprio carrinho	C08 - Usuário tentando acessar carrinho de outro usuário	O teste valida que usuários não podem acessar	1. Criar carrinho com usuário A

		carrinhos de outros usuários.	2. Tentar GET `/carrinhos/:id` autenticado como usuário B 3. Validar status 403 ou erro de autorização