US003 - Produtos

Sendo um vendedor de uma loja com cadastro já realizado
Gostaria de poder me autenticar e cadastrar produtos no Marketplace do ServeRest
Para poder cadastrar, editar, atualizar e excluir meus produtos

DoR

- Banco de dados e infraestrutura para desenvolvimento disponibilizados;
- API de cadastro de usuários implementada;
- API de autenticação implementada;
- Ambiente de testes disponibilizado.

DoD

- CRUD de cadastro de Produtos implementado (CRIAR, ATUALIZAR, LISTAR E DELETAR);
- Análise de testes cobrindo a rota de produtos;
- Matriz de rastreabilidade atualizada;
- Automação de testes baseado na análise realizada;

Acceptance Criteria

- Usuários não autenticados não devem conseguir realizar ações na rota de Produtos;
- Não deve ser possível realizar o cadastro de produtos com nomes já utilizados;
- Não deve ser possível excluir produtos que estão dentro de carrinhos (dependência API Carrinhos);
- Caso não exista produto com o o ID informado na hora do UPDATE, um novo produto deverá ser criado;
- Produtos criados através do PUT não poderão ter nomes previamente cadastrados;
- Os testes executados deverão conter evidências;
- A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos.

Cenários de Teste

CT-025 – Cadastro de produto com dados válidos

Pré-condição: Usuário autenticado (token válido)

Passos:
Realizar POST `/produtos` com o seguinte payload:

```
{
  "nome": "Camiseta Tech",
  "preco": 99,
  "descricao": "Camiseta com estampa tecnológica",
  "quantidade": 10
}
```

Resultado Esperado:

- Status code `201 Created`
- Mensagem: "Cadastro realizado com sucesso"

- ID do produto retornado

CT-026 – Cadastro de produto sem autenticação

Pré-condição: Nenhum token fornecido ou token inválido

Passos:
Realizar POST `/produtos` com payload válido, sem header Authorization, como:

```
{
  "nome": "Fone Bluetooth",
  "preco": 199.99,
  "descricao": "Fone com cancelamento de ruído",
  "quantidade": 5
}
```

Resultado Esperado:

- Status code `401 Unauthorized`
- Mensagem indicando necessidade de autenticação

CT-027 – Cadastro com nome já utilizado

Pré-condição: Produto "Camiseta Tech" já cadastrado

Passos:
Realizar POST `/produtos` com o mesmo nome e dados similares

Resultado Esperado:

- Status code `400`
- Mensagem: "`Já existe produto com esse nome`"

CT-028 – Edição de produto com dados válidos

Pré-condição: Produto já cadastrado + usuário autenticado

Passos:
Realizar PUT `/produtos/{id}` com payload atualizando preço e descrição, como:

```
{
  "nome": "Camiseta Tech",
  "preco": 89.90,
  "descricao": "Camiseta tech com novo design",
  "quantidade": 10
}
```

Resultado Esperado:

- Status code `200`
- Mensagem: "`Alterado com sucesso`"

CT-029 – PUT em produto com ID inexistente (criação implícita)

Pré-condição: Nenhum produto com ID `xyz123`

Passos:
Realizar PUT `/produtos/xyz123` com payload válido

Resultado Esperado:

- Status code `200` (conforme Swagger)
- Produto é criado com novo ID
- Mensagem coerente com criação.

CT-030 – PUT com nome duplicado

Pré-condição: Dois produtos existentes com nomes iguais.

Passos:
Atualizar um dos produtos trocando seu nome para o nome de outro já existente, como:

```
{
  "nome": "Fone Bluetooth",  // já existente
  "preco": 189.90,
  "descricao": "Versão atualizada",
  "quantidade": 10
}
```

Resultado Esperado:

- Status code `400`
- Mensagem: "`Já existe produto com esse nome`"

CT-031 – Deletar produto sem estar vinculado a carrinho

Pré-condição: Produto não vinculado a nenhum carrinho

Passos:
Realizar DELETE `/produtos/{id}` com token válido

Resultado Esperado:

- Status code `200`
- Mensagem: "`Registro excluído com sucesso`"

CT-032 – Deletar produto vinculado a carrinho

Pré-condição: Produto está em um carrinho ativo

Passos:
Realizar DELETE `/produtos/{id}`

Resultado Esperado:

- Status code `400`
- Mensagem: "`Produto faz parte de carrinho`"

CT-033 – Listar todos os produtos

Pré-condição: Existem produtos cadastrados

Passos:
Realizar GET `/produtos`

Resultado Esperado:

- Status code `200`
- Lista com produtos cadastrados

CT-034 – Criar produto com campo obrigatório ausente

Passos:
Realizar POST `/produtos` sem o campo `"quantidade"`, como:

```
{
  "nome": "Controle Gamer",
  "preco": 150.00,
  "descricao": "Controle sem fio para jogos"
}
```

Resultado Esperado:

- Status code `400`
- Mensagem indicando ausência de campo obrigatório

CT-035 – Criar produto com quantidade negativa

Passos:
Realizar POST `/produtos` com `"quantidade": -5`

Resultado Esperado:

- Status code `400`
- Mensagem indicando quantidade incoerente.

CT-036 – Criar produto com preço zero

Passos:
Realizar POST `/produtos` com `"preco": 0`

Resultado Esperado:

- Status code `400` (se não permitido)
- Ou `201 Cadastrado com sucesso` (se permitido)

CT-037 – Criar produto com nome em branco

Passos:
Realizar POST `/produtos` com `"nome": ""`

Resultado Esperado:

- Status code `400`
- Mensagem indicando nome obrigatório

CT-038 – Tentar listar produtos com token inválido

Pré-condição: Token inválido no header

Passos:
Realizar GET `/produtos` com token incorreto

Resultado Esperado:

- Status code `401 Unauthorized`
- Mensagem coerente de rejeição

CT-039 – PUT com campos extras inesperados

Pré-condição: Produto existente

Passos:
Enviar PUT `/produtos/{id}` com campos irrelevantes no payload

Resultado Esperado:

- `200 OK` (se a API ignorar os campos extras)
- Ou `400 Bad Request` (se validar estrutura)

Priorização dos Testes

ID	Endpoint	Cenário	Tipo	Prioridade	Status
CT-025	`/produtos`	Cadastro de produto com dados válidos	Funcional / Automatizado	Alta	NÃO EXECUTADO
CT-026	`/produtos`	Cadastro de produto sem autenticação	Segurança / Funcional	Alta	EXECUTADO
CT-027	`/produtos`	Cadastro com nome de produto já existente	Tabela de Decisão / Funcional	Alta	NÃO EXECUTADO
CT-028	`/produtos/{id}`	PUT em produto para alterar nome/preço	Funcional / Automatizado	Alta	NÃO EXECUTADO
CT-029	`/produtos/{id}`	PUT com ID inexistente (criação implícita)	Funcional / Automatizado	Média	NÃO EXECUTADO
CT-030	`/produtos/{id}`	PUT com nome de outro produto (duplicado)	Tabela de Decisão / Funcional	Alta	NÃO EXECUTADO
CT-031	`/produtos/{id}`	DELETE de produto não vinculado a carrinho	Funcional / Automatizado	Alta	NÃO EXECUTADO
CT-032	`/produtos/{id}`	DELETE de produto vinculado a carrinho	Segurança / Regras de Negócio	Crítica	NÃO EXECUTADO

CT-033	`/produtos`	GET listar todos os produtos	Funcional Manual	Baixa	NÃO EXECUTADO
CT-034	`/produtos`	POST sem campo obrigatório (quantidade)	Particionamento / Funcional	Alta	NÃO EXECUTADO
CT-035	`/produtos`	POST com quantidade negativa	Valor Limite / Funcional	Alta	NÃO EXECUTADO
CT-036	`/produtos`	POST com preço igual a zero	Valor Limite / Ambiguidade	Média	NÃO EXECUTADO
CT-037	`/produtos`	POST com nome vazio	Particionamento / Funcional	Alta	NÃO EXECUTADO
CT-038	`/produtos`	GET com token inválido	Segurança	Alta	NÃO EXECUTADO
CT-039	`/produtos/{id}`	PUT com campos adicionais não documentados	Exploratória / Estrutura	Média	NÃO EXECUTADO