

US001 - Usuários

Sendo um vendedor de uma loja

Gostaria de poder me cadastrar no Marketplace do ServeRest

Para poder realizar as vendas dos meus produtos

DoR

- Banco de dados e infraestrutura para desenvolvimento disponibilizados;
- Ambiente de testes disponibilizado.

DoD

- CRUD de cadastro de vendedores (usuários) implementado (CRIAR, ATUALIZAR, LISTAR E DELETAR);
- Análise de testes cobrindo todos verbos;
- Matriz de rastreabilidade atualizada;
- Automação de testes baseado na análise realizada;

Acceptance Criteria

- Os vendedores (usuários) deverão possuir os campos NOME, E-MAIL, PASSWORD e ADMINISTRADOR;
- Não deverá ser possível fazer ações e chamadas para usuários inexistentes;
 - Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado;
- Não deve ser possível criar um usuário com e-mail já utilizado;
- Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT;
- Os testes executados deverão conter evidências;
- Não deverá ser possível cadastrar usuários com e-mails de provedor gmail e hotmail;
- Os e-mails devem seguir um padrão válido de e-mail para o cadastro;
- As senhas devem possuir no mínimo 5 caracteres e no máximo 10 caracteres;
- A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos.

Cenários de Teste [🔗](#)

CT-001 – Cadastro válido de novo usuário [🔗](#)

Pré-condição: Nenhum usuário cadastrado com o e-mail `joao@valido.com`

Passos:

- Realizar uma requisição `POST /usuarios`
- Enviar o seguinte payload:

```
1 {  
2   "nome": "João Silva",  
3   "email": "joao@valido.com",  
4   "password": "123456",  
5   "administrador": "true"  
6 }
```

- **Resultado Esperado:**

- Status code 201 Created
 - Corpo da resposta contém a mensagem: "Cadastro realizado com sucesso"
 - ID do usuário é retornado
-

CT-002 – Cadastro com e-mail duplicado [↗](#)

- **Pré-condição:** Já existe um usuário com o e-mail joao@valido.com
 - **Passos:**
 - a. Realizar um POST /usuarios com o mesmo e-mail do usuário existente
 - **Resultado Esperado:**
 - Status code 400 Bad Request
 - Mensagem: "Este email já está sendo usado"
-

CT-003 – Cadastro com e-mail inválido [↗](#)

- **Passos:**
 - a. Enviar POST /usuarios com email no formato inválido: "joao@val*d.com"
 - **Resultado Esperado:**
 - Status code 400
 - Mensagem de erro indicando formato inválido
-

CT-004 – Cadastro com e-mail de domínio restrito (Gmail) [↗](#)

- **Passos:**
 - a. Enviar POST /usuarios com email "joao@gmail.com"
 - **Resultado Esperado:**
 - Status code 400
 - Mensagem de erro indicando que domínios gmail/hotmail não são aceitos
-

CT-005 – Cadastro com senha muito curta (<5 caracteres) [↗](#)

- **Passos:**
 - a. Enviar POST /usuarios com senha "123"
 - **Resultado Esperado:**
 - Status code 400
 - Mensagem: "A senha deve conter entre 5 e 10 caracteres"
-

CT-006 – Cadastro com senha longa demais (>10 caracteres) [↗](#)

- **Passos:**
 - a. Enviar POST /usuarios com senha "12345678910"
 - **Resultado Esperado:**
 - Status code 400
 - Mensagem: "A senha deve conter entre 5 e 10 caracteres"
-

CT-007 – GET em usuário inexistente [🔗](#)

- **Pré-condição:** ID inexistente na base
 - **Passos:**
 - a. Realizar `GET /usuarios/{id-invalido}`
 - **Resultado Esperado:**
 - Status code `400` ou `404`
 - Mensagem: `"Usuário não encontrado"`
-

CT-008 – PUT criando novo usuário com ID inexistente [🔗](#)

- **Pré-condição:** Nenhum usuário com ID `xyz123`
 - **Passos:**
 - a. Enviar `PUT /usuarios/xyz123` com payload de novo usuário
 - **Resultado Esperado (⚠️ Ambíguo):**
 - A definir: *cria novo usuário ou retorna erro?*
-

CT-009 – PUT em usuário com e-mail duplicado [🔗](#)

- **Pré-condição:** Dois usuários já cadastrados com e-mails distintos
 - **Passos:**
 - a. Tentar editar o segundo usuário usando o e-mail do primeiro
 - **Resultado Esperado:**
 - Status code `400`
 - Mensagem: `"Este email já está sendo usado"`
-

CT-010 – DELETE de usuário inexistente [🔗](#)

- **Pré-condição:** ID inválido
- **Passos:**
 - a. Realizar `DELETE /usuarios/{id-invalido}`
- **Resultado Esperado:**
 - Status code `200` ou `400`
 - Mensagem coerente: `"Nenhum registro excluído"` ou `"Usuário não encontrado"`

Observação: Swagger permite 200 com `"Nenhum registro excluído"`, porém esse retorno pode ser confuso se ID for inválido.

CT-011 – GET todos os usuários [🔗](#)

- **Pré-condição:** Existem usuários cadastrados
- **Passos:**
 - a. Realizar `GET /usuarios`
- **Resultado Esperado:**
 - Status code `200 OK`
 - Corpo da resposta com array de usuários válidos

CT-012 – Verificar campos obrigatórios [🔗](#)

- **Passos:**
 - a. Enviar `POST /usuarios` sem o campo "email"
- **Resultado Esperado:**
 - Status code `400`
 - Mensagem de campo obrigatório ausente

CT-013 – Campo “administrador” com valor inválido [🔗](#)

- **Passos:**
 - a. Enviar `"administrador": "talvez"` (em vez de "true" ou "false")
- **Resultado Esperado:**
 - Status code `400`
 - Mensagem de erro validando o tipo

CT-014 – Criação de usuário com nome vazio [🔗](#)

- **Passos:**
 - a. Enviar payload com `"nome": ""`
- **Resultado Esperado:**
 - Status code `400`
 - Mensagem: `"Nome é obrigatório"`

Priorização dos Testes [🔗](#)

ID	Endpoint	Cenário	Tipo	Prioridade	Status
CT-001	/usuarios	Cadastro de novo usuário com dados válidos	Funcional Manual	Alta	EXECUTADO
CT-002	/usuarios	Cadastro com e-mail já existente	Funcional Manual	Alta	EXECUTADO
CT-003	/usuarios	Cadastro com e-mail inválido	Funcional Manual	Média	EXECUTADO
CT-004	/usuarios	Cadastro com domínio de e-mail restrito (gmail/hotmail)	Funcional Manual	Média	EXECUTADO
CT-005	/usuarios	Cadastro com senha curta (< 5 caracteres)	Valor Limite / Funcional	Alta	EXECUTADO
CT-006	/usuarios	Cadastro com senha longa (> 10 caracteres)	Valor Limite / Funcional	Alta	EXECUTADO
CT-007	/usuarios/{id}	GET em usuário inexistente	Funcional Manual	Média	EXECUTADO
CT-008	/usuarios/{id}	PUT com ID inexistente cria novo usuário	Funcional Manual	Média	BLOQUEADO

CT-009	/usuarios/{id}	PUT com e-mail duplicado	Tabela de Decisão / Funcional	Alta	EXECUTADO
CT-010	/usuarios/{id}	DELETE de usuário inexistente	Funcional Manual	Média	EXECUTADO
CT-011	/usuarios	GET para listar todos os usuários	Funcional Manual	Baixa	EXECUTADO
CT-012	/usuarios	Cadastro com campo obrigatório ausente (e.g., e- mail)	Particionamento / Funcional	Alta	EXECUTADO
CT-013	/usuarios	Campo administrador com valor inválido	Particionamento / Funcional	Média	EXECUTADO
CT-014	/usuarios	Cadastro com nome em branco	Particionamento / Funcional	Alta	EXECUTADO