US-User [API] (Cenários detalhados)

User store

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;

Critério de aceite US001

Critério de aceite 01	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do teste
Os vendedores (usuários) deverão possuir os campos NOME, E-MAIL, PASSWORD e ADMINISTRADOR.	**ID:** US001-CT01 **Cenário de Teste 1:** Criação de usuário com todos os campos obrigatórios corretamente preenchidos Objetivo: Validar que o sistema permite criar um usuário quando todos os campos obrigatórios são fornecidos corretamente. **Pré-condições**: A API de cadastro está disponível.	Enviar uma requisição `POST` para o endpoint `/api/usuarios` com os seguintes campos: **nome:** "Carlos Silva" **email:** "carlos@empresa.com" **password:** "SenhaForte123" **administrador:** "true"	Código de status: 201 Mensagem de sucesso e retorno dos dados do usuário.	**"Obrigatório (Must)"**	**Não Executado**
	ID: US001-CT02 **Cenário de Teste 2**: Tentativa de criação de usuário sem o campo NOME **Objetivo**: Validar que o sistema rejeita a criação	**Passos**: Enviar um `POST` para `/api/usuarios` omitindo o campo `nome`. Verificar se a resposta retorna **400 Bad Request** com mensagem apropriada.	Código: 400 Mensagem: `"nome é obrigatório"`	**"Obrigatório (Must)"**	**Não Executado**

	de usuário sem o campo `nome`. **Pré-condições**: A API de cadastro está disponível.				
	ID: US001-CT03 **Cenário de Teste 3: Tentativa de criação de usuário sem o campo E-MAIL** **Objetivo**: Validar que o sistema exige o campo `email` no cadastro de usuário.	**Passos**: -Enviar um `POST` omitindo o campo `email`. -Verificar resposta com erro.	Código: 400 Mensagem: `"email é obrigatório"`	**"Obrigatório (Must)"**	**Não Executado**
	ID: US001-CT04 **Cenário de Teste 4:** Tentativa de criação de usuário sem o campo PASSWORD **Objetivo**: Verificar se a API rejeita usuários sem senha.	**Passos**: Enviar um `POST` omitindo o campo `password`. -Verificar se retorna erro 400.	Código: 400 Mensagem: `"password é obrigatório"`	**"Obrigatório (Must)"**	**Não Executado**

	ID: US001-CT05
	
	Cenário de Teste 5: Tentativa de criação de usuário sem o campo ADMINISTRADOR
	
	Objetivo: Garantir que o campo `administrador` é obrigatório e validado.
		`Passos:`
		
		`Enviar um POST omitindo o campo administrador.`
		
		`-Verificar se o sistema responde com erro.`
			Código: 400
			
			Mensagem: `"administrador é obrigatório"`
				Importante (Should).	**Não Executado**

	ID: US001-CT06 **Cenário de Teste 6: Tentativa de criação de usuário com tipo inválido no campo ADMINISTRADOR** **Objetivo**: Verificar se o sistema valida os tipos de dados do campo `administrador`.	**Passos**: Enviar `POST` com `administrador: 123` (valor numérico inválido).	Código: 400 Mensagem: `"administrador deve ser 'true' ou 'false'"`	**Importante (Should).**	**Não Executado**
	ID: US001-CT07 **Cenário de Teste Extra:** Login com usuário recém-cadastrado **Objetivo:** Verificar se o usuário criado com	**Passos**: Enviar uma requisição `POST` para `/api/login` com: **email:** "carlos@empresa.com"	Código: `200` Corpo da resposta contém: `{ token: "<jwt_token>" }`	**Obrigatória (Must).**	**Não Executado**

	sucesso consegue autenticar com seu e-mail e senha. Pré-condições: Usuário válido cadastrado. **Pré-condições:** Usuário válido cadastrado.	**password:** "SenhaForte123"			

Critério de aceite US002

Critério de aceite 02	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do teste
Não deverá ser possível fazer ações e chamadas para usuários inexistentes;	**ID:** US002-CT01 **Cenário de Teste 1: Login com usuário inexistente** **Objetivo:** Garantir que o sistema rejeita autenticação de um usuário que não está cadastrado. **Pré-condição:** Nenhum usuário com o e-mail testado existe no banco.	Enviar uma requisição POST para o endpoint `/api/login` com: **email:** `naoexiste@empresa.com` **password:** `QualquerSenha123` -Verificar se a resposta retorna 401 Unauthorized -Verificar se a mensagem de erro informa "Usuário e/ou senha inválidos"	Código: 401 Mensagem: "Usuário e/ou senha inválidos"	**Obrigatória (Must).**	**Não Executado**
	ID: US002-CT02 **Cenário de Teste 2: Consulta de usuário inexistente por ID** **Objetivo:** Validar que não é possível consultar um usuário que não existe. **Pré-condição:** O ID consultado não está no banco.	Enviar uma requisição GET para `/api/usuarios/9999` (ID inexistente) Verificar se a resposta retorna 404 Not Found Verificar se a mensagem informa "Usuário não encontrado"	Código: 404 Mensagem: "Usuário não encontrado"	**Importante (Should).**	**Não Executado**
	ID: US002-CT03 **Cenário de Teste 3: Atualização de usuário inexistente** **Objetivo:** Garantir que o sistema não permite atualização de usuários	Enviar uma requisição PUT para `/api/usuarios/9999` com dados válidos Verificar se retorna 404 Not Found	Código: 404 Mensagem: "Usuário não encontrado"	**Importante (Should).**	**Não Executado**

	que não existem. **Pré-condição:** O ID informado na requisição não existe no banco.	Verificar se a mensagem informa "Usuário não encontrado"			

	ID: US002-CT04
	Cenário de Teste 4: Exclusão de usuário inexistente
	Objetivo: Verificar que não é possível deletar um usuário que não existe.
	Pré-condição: O ID fornecido não está cadastrado.
		`Enviar uma requisição DELETE para /api/usuarios/9999`
		`Verificar se retorna 404 Not Found`
		`Verificar se a mensagem informa "Usuário não encontrado"`
			Código: 400
			Mensagem: "Este e-mail já está sendo utilizado"
				Importante (Should).	**Não Executado**

Critério de aceite US003

Critério de aceite 03	Cenários de Teste	Entrada	Resultado Esperado		Status do teste
Não deve ser possível criar um usuário com e-mail já utilizado;	**ID:** US003-CT01 **Cenário de Teste 1: Tentativa de criar usuário com e-mail já cadastrado** **Objetivo:** Verificar que a API impede a criação de um novo usuário com um e-mail já registrado. **Pré-condições:** Um usuário já existe com o e-mail `joao@empresa.com`	Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** `João da Silva` **email:** `joao@empresa.com` **password:** `SenhaSegura123` **administrador:** `false` Verificar se a resposta retorna 400 Bad Request Verificar se a mensagem de erro informa que o e-mail já está em uso	Código: 400 Mensagem: "Este e-mail já está sendo utilizado"	**Obrigatória (Must).**	**Não Executado**

	ID: US003-CT02
	Cenário de Teste 2: Criar usuário com e-mail diferente após tentativa inválida
	Objetivo: Garantir que é possível criar um usuário com um novo e-mail mesmo após uma tentativa
		`Enviar uma requisição POST para /api/usuarios com:`
		`nome: João da Silva`
		`email: joao2@empresa.com`
		`password: SenhaSegura123`
			Código: 201
			Mensagem: Sucesso na criação do usuário
				Importante (Should).	**Não Executado**

	anterior com e-mail repetido.
	Pré-condições: O e-mail `joao@empresa.com` já está cadastrado
		`administrador: false`
		`Verificar se a resposta retorna 201 Created`
		`Verificar se o usuário foi criado com um novo ID`			

Critério de aceite US004

Critério de aceite 04	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do teste

Caso não seja encontrado usuário com o ID informado no PUT, um novo usuário deverá ser criado;	**ID:** US004-CT01
	Cenário de Teste 1: PUT com ID inexistente deve criar novo usuário
	Objetivo: Verificar que o sistema cria um novo usuário quando o ID passado no PUT não existe.
	Pré-condições: O ID `9999` não existe no sistema
		`Enviar uma requisição PUT para /api/usuarios/9999 com:`
		`nome: Marina Souza`
		`email: marina@empresa.com`
		`password: Senha123`
		`administrador: false`
		`Verificar se a resposta retorna 201 Created`
		`Verificar se um novo usuário foi criado com novo ID`
			Código: 201
			Mensagem: "Usuário criado com sucesso"
			Corpo da resposta deve conter os dados do novo usuário
				Obrigatório (Must)	**Não Executado**

	ID: US004-CT02 **Cenário de Teste 2: PUT com ID existente deve atualizar o usuário** **Objetivo:** Garantir que o sistema atualiza o usuário se o ID informado já existir. **Pré-condições:** Um usuário com ID `5` já está cadastrado	Enviar uma requisição PUT para `/api/usuarios/5` com dados atualizados: **nome:** `Marina Atualizada` **email:** `marina@empresa.com` **password:** `NovaSenha123` **administrador:** `true` Verificar se a resposta retorna 200 OK Verificar se os dados do usuário foram	Código: 200 Mensagem: "Usuário atualizado com sucesso" Corpo da resposta com os dados atualizados	**Obrigatório (Must)**	**Não Executado**

		atualizados			

Critério de aceite US005

Critério de aceite 05	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do teste
Não deve ser possível cadastrar usuário com e-mail já utilizado utilizando PUT;	**ID:** US005-CT01 **Cenário de Teste 1: Tentativa de atualizar usuário com e-mail já utilizado por outro usuário** **Objetivo:** Garantir que a API rejeita a atualização de um usuário para um e-mail já usado por outro. **Pré-condições:** Usuário com ID `1` cadastrado com e-mail `joao@empresa.com` Usuário com ID `2` cadastrado com e-mail `maria@empresa.com`	Enviar uma requisição PUT para `/api/usuarios/2` com: **nome:** `Maria Almeida` **email:** `joao@empresa.com` **password:** `SenhaNova123` **administrador:** `false` Verificar se a resposta retorna 400 Bad Request Verificar se a mensagem informa que o e-mail já está em uso	Código: 400 Mensagem: "Este e-mail já está sendo utilizado por outro usuário"	**Obrigatório (Must)**	**Não Executado**
	ID: US005-CT02 **Cenário de Teste 2: PUT com o mesmo e-mail do próprio usuário (sem alteração)** **Objetivo:** Garantir que o usuário pode atualizar seus dados sem trocar o e-mail, mantendo o e-mail atual. **Pré-condições:** Usuário com ID `1` cadastrado com e-mail `joao@empresa.com`	Enviar uma requisição PUT para `/api/usuarios/1` com: **nome:** `João Atualizado` **email:** `joao@empresa.com` **password:** `SenhaAtualizada123` **administrador:** `true` Verificar se a resposta retorna 200 OK Verificar se os dados foram atualizados com sucesso	Código: 200 Mensagem: "Usuário atualizado com sucesso" Corpo da resposta com os novos dados (exceto o e-mail, que permanece)	**Importante (Should)**	**Não Executado**

Critério de aceite US006

Critério de aceite 06	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do

					teste
Os testes executados deverão conter evidências;	**ID:** US006-CT01 **Cenário de Teste 1: Registro de evidência para teste bem-sucedido** **Objetivo:** Garantir que, ao executar um teste com sucesso, uma evidência (como log, print, ou relatório) seja gerada. **Pré-condições:** Ambiente de testes automatizado e configurado Execução de testes via ferramenta (ex: Postman, JMeter, Cypress, ou Jest)	Passos: Executar um teste positivo (ex: cadastro de usuário com dados válidos) Verificar se foi gerado arquivo de evidência contendo: -Request enviada -Response recebida -Status code -Data/hora da execução -Resultado (Aprovado)	Evidência salva automaticamente em diretório definido (ex: `/evidencias/testes_usuarios/`) Nome do arquivo contendo o ID do teste e data	**Obrigatório (Must)**	**Não Executado**
	ID: US006-CT02 **Cenário de teste 2:** Captura de evidência ao ocorrer uma falha em teste automatizado **Objetivo:** Garantir que falhas em testes também gerem evidências completas para análise. **Pré-condições:** Ambiente de testes automatizado configurado para capturar falhas	Passos: • Executar um teste negativo (ex: tentativa de criar usuário com e-mail duplicado) • Verificar se a evidência inclui: -Request enviada -Response recebida -Status code -Mensagem de erro -Stack trace (se aplicável)	Evidência armazenada Nome do arquivo indicando que foi uma falha.	**Obrigatório (Must)**	**Não Executado**
	ID: US006-CT03 **Cenário de Teste 3: Evidência salva com print de tela (para testes manuais)** **Objetivo:** Garantir que testes manuais possuam screenshots como evidência. **Pré-condições:** Testes sendo executados manualmente	Passos: • Realizar o teste manualmente (ex: via frontend) • Capturar print de tela da operação final • Salvar o print com o ID do caso de teste • Associar o print ao caso de teste no plano de testes	Print salvo Associado corretamente no relatório/manual de testes	**Importante (Should)**	**Não Executado**

Critério de aceite US007

Critério de aceite 07	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do teste
Não deverá ser possível cadastrar usuários com e-mails de provedor gmail e hotmail;	**ID:** US007-CT01 **Cenário de Teste 1: Tentativa de cadastro com e-mail do Gmail** **Objetivo:** Verificar se o sistema bloqueia o cadastro de usuários com e-mail do provedor Gmail. **Pré-condições:** • API de cadastro de usuário disponível.	Passos: Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** "Ana Silva" **email:** "ana@gmail.com" **password:** "Senha123" **administrador**: "false" Verificar a resposta da API.	Código: 400 Mensagem: "Não é permitido o uso de e-mails do provedor gmail."	**Obrigatório (Must)**	**Não Executado**
	ID: US007-CT02 **Cenário de Teste:** Tentativa de cadastro com e-mail do provedor Hotmail. **Objetivo:** Verificar se o sistema bloqueia o cadastro de usuários com e-mail do provedor Hotmail. **Pré-condições:** API de cadastro de usuário disponível.	Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** "Bruno Costa" **email:** "bruno@hotmail.com" **password:** "Senha123" **administrador:** "true" Verificar a resposta da API.	Código: 400 Mensagem: "Não é permitido o uso de e-mails do provedor hotmail."	**Obrigatório (Must)**	**Não Executado**
	ID: US007-CT03 **Cenário de Teste:** Cadastro de usuário com e-mail de domínio permitido. **Objetivo:** Garantir que usuários com e-mails de outros provedores (ex: empresa.com) sejam cadastrados normalmente. **Pré-condições:** API de cadastro disponível.	Passos: Enviar um POST para `/api/usuarios` com e-mail "joao@empresa.com". Verificar a resposta.	Código: 201 Usuário criado com sucesso.	**Obrigatório (Must)**	**Não Executado**

Critério de aceite US008

Critério de aceite 08	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status do

					teste
Os e-mails devem seguir um padrão válido de e-mail para o cadastro;	**ID:** US008-CT01 **Cenário de Teste 1: Cadastro com e-mail válido** **Objetivo:** Garantir que o sistema aceite e-mails válidos no formato correto. **Pré-condições:** API de cadastro disponível.	Passos: Enviar uma requisição POST para `/api/usuarios` com: **nome:** "Lucas Lima" **email:** "lucas@empresa.com" **password:** "SenhaSegura123" **administrador:** "true" Verificar a resposta.	Código: 201 Usuário cadastrado com sucesso.	**Obrigatório (Must)**	**Não Executado**
	ID: US008-CT02 **Cenário de Teste 2: Cadastro com e-mail sem o caractere @** **Objetivo:** Verificar se o sistema rejeita e-mails sem o caractere "@". **Pré-condições:** API de cadastro disponível.	Passos: Enviar uma requisição POST para `/api/usuarios` com o e-mail: "joaoempresa.com". Verificar a resposta.	Código: 400 Mensagem: "Formato de e-mail inválido."	**Obrigatório (Must)**	**Não Executado**
	ID: US008-CT03 **Cenário de Teste 3: Cadastro com e-mail sem domínio** **Objetivo:** Verificar se o sistema rejeita e-mails sem domínio. **Pré-condições:** API de cadastro disponível.	Passos: Enviar um POST com o e-mail: "joana@". Verificar a resposta.	Código: 400 Mensagem: "Formato de e-mail inválido."	**Obrigatório (Must)**	**Não Executado**
	ID: US008-CT04 **Cenário de Teste 4: Cadastro com e-mail sem nome de usuário** **Objetivo:** Validar a rejeição de e-mails sem nome de usuário antes do @. **Pré-condições:** API de cadastro disponível.	Passos: Enviar um POST com o e-mail: "@empresa.com". Verificar a resposta.	Código: 400 Mensagem: "Formato de e-mail inválido."	**Obrigatório (Must)**	**Não Executado**

Critério de aceite US009

Critério de aceite 09	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Status

					teste
As senhas devem possuír no mínimo 5 caracteres e no máximo 10 caracteres;	**ID:** US009-CT01 **Cenário de Teste 1: Cadastro com senha dentro do limite de caracteres (5 caracteres)** **Objetivo:** Verificar se o sistema aceita senhas com 5 caracteres. **Pré-condições:** API de cadastro disponível.	Passos: Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** "Carlos Souza" **email:** "carlos@empresa.com" **password:** "12345" **administrador:** "false" Verificar a resposta.	Código: 201 Usuário cadastrado com sucesso.	**Obrigatório (Must)**	**Não Executado**
	ID: US009-CT02 **Cenário de Teste 2: Cadastro com senha abaixo do limite de caracteres (menos de 5 caracteres)** **Objetivo:** Verificar se o sistema rejeita senhas com menos de 5 caracteres. **Pré-condições:** API de cadastro disponível.	Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** "Renato Lima" **email:** "renato@empresa.com" **password:** "123" **administrador:** "true" Verificar a resposta.	Código: 400 Mensagem: "A senha deve ter entre 5 e 10 caracteres."	**Obrigatório (Must)**	**Não Executado**
	ID: US009-CT03 **Cenário de Teste 3: Cadastro com senha no limite máximo de caracteres (10 caracteres)** **Objetivo:** Verificar se o sistema aceita senhas com 10 caracteres. Pré-condições: API de cadastro disponível.	Passos: Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** "Felipe Almeida" **email:** "felipe@empresa.com" **password:** "senha12345" **administrador:** "false" Verificar a resposta.	Código: 201 Usuário cadastrado com sucesso.	**Obrigatório (Must)**	**Não Executado**
	ID: US009-CT04 **Cenário de Teste 4: Cadastro com senha acima do limite de caracteres (mais de 10 caracteres)** **Objetivo:** Verificar se o sistema rejeita senhas com mais de 10 caracteres. **Pré-condições:** API de cadastro disponível.	Passos: Enviar uma requisição POST para `/api/usuarios` com os seguintes dados: **nome:** "Juliana Costa" **email:** "juliana@empresa.com" **password:** "senha12345678" **administrador:** "true"	Código: 400 Mensagem: "A senha deve ter entre 5 e 10 caracteres."	**Obrigatório (Must)**	**Não Executado**

		Verificar a resposta.			

Critério de aceite US010

Critério de aceite 10	Cenários de Teste	Entrada	Resultado Esperado	Prioridade	Stastus do teste
A cobertura de testes deve se basear no Swagger e ir além, cobrindo cenários alternativos.	**ID:** US010-CT01 **Cenário de Teste 1: Teste de cenários alternativos para cadastro de usuário** **Objetivo:** Validar que o sistema lida com cenários alternativos (erro de validação, dados inválidos, etc.), além do que está documentado no Swagger. Pré-condições: -API de cadastro disponível. -Swagger da API acessível.	**Passos:** Testar o endpoint POST `/api/usuarios` enviando dados inválidos para os campos obrigatórios. Exemplo: Enviar um e-mail inválido, senha muito curta, ou valores inválidos para os campos. Testar o endpoint PUT `/api/usuarios/{id}` com ID inválido ou com dados inconsistentes. Testar o endpoint DELETE `/api/usuarios/{id}` tentando deletar um usuário inexistente.	Para o cadastro de usuário com dados inválidos, o código de resposta deve ser 400 (Bad Request). Para o PUT e DELETE com ID inválido, o código de resposta deve ser 404 (Not Found). Mensagens de erro claras e apropriadas, como "E-mail inválido", "ID não encontrado", etc.	**Obrigatório (Must)**	**Não Executado**
	ID: US010-CT02 **Cenário de Teste 2: Teste de autenticação e autorização** **Objetivo:** Verificar a autenticação e autorização da API, validando que as permissões estão sendo corretamente aplicadas conforme documentado no Swagger e além. **Pré-condições:** API de login e cadastro de usuários disponíveis. Swagger da API acessível.	Passos: Realizar um login com um usuário válido e verificar se o token é retornado corretamente. Tentar acessar um endpoint restrito (por exemplo, listagem de usuários) sem autenticação e verificar se a resposta é 401 (Unauthorized). Tentar acessar o mesmo endpoint com um token válido e verificar se o acesso é permitido.	A resposta para um login válido deve ser 200 (OK) com o token Bearer. A resposta para um acesso sem autenticação deve ser 401 (Unauthorized). A resposta para um acesso com token válido deve ser 200 (OK).	**Obrigatório (Must)**	**Não Executado**

	ID: US010-CT03 **Cenário de Teste 3: Teste de limites e performance (com base no Swagger e além)** **Objetivo:** Validar o desempenho e limites da API, incluindo testes de carga e performance. Pré-condições: API de cadastro disponível. Swagger da API acessível.	Passos: Enviar um grande volume de requisições POST para `/api/usuarios` com dados válidos e validar se o sistema consegue processá-las sem erros. Enviar uma requisição POST para `/api/usuarios` com dados válidos repetidamente (stress test) para verificar como o sistema se comporta sob carga. Testar limites como tamanho de campos (por exemplo, o tamanho máximo do campo "nome").	O sistema deve processar as requisições corretamente sem falhas durante o teste de carga. Respostas dentro de um tempo aceitável (ex.: <2 segundos por requisição). O sistema deve retornar mensagens de erro apropriadas quando os limites de dados forem excedidos.	**Importante (Should)**	**Não Executado**