US-USER [API] Relatório de teste

Critério de aceite US001 🛭

ID	Caso de Teste	Ti po M/ A	Prioridade	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status	Observ ações
ID: US001- CT01	Criação de usuário com todos os campos obrigatórios corretament e preenchidos Objetivo: Validar que o sistema permite criar um usuário quando todos os campos obrigatórios são fornecidos corretamente .	A	(MUST)	Pré- condições: A API de cadastro está disponível.	Enviar uma requisição POST para o endpoint /api/usua rios com os seguintes campos: nome: "Carlos Silva" email: "carlos@e mpresa.co m" password: "SenhaFort e123" administra dor: "true"	Código de status: 201 Mensagem de sucesso e retorno dos dados do usuário.	✓ Status code é 201 - Usuário criado com sucesso ✓ Mensagem de sucesso está presente ✓ Resposta contém ID do usuário	✓ PASSED	
ID: US001- CT02	Tentativa de criação de usuário sem o campo NOME Objetivo: Validar que o sistema rejeita a criação de usuário sem o campo nome .	Α	(MUST)	Pré- condições: A API de cadastro está disponível.	Enviar um POST para /api/usua rios omitindo o campo nome . Verificar se a resposta retorna 400 Bad Request com mensagem apropriada .	Código: 400 Mensagem : "nome é obrigatór io"	✓ Resposta de erro com status 400 ✓ Mensagem de erro deve ser JSON ✓ Mensagem de erro correta	✓ PASSED	

ID: US001- CT03	Tentativa de criação de usuário sem o campo E-MAIL Objetivo: Validar que o sistema exige o campo email no cadastro de usuário.	Α	(MUST)	"API ativa; nome e senha válidos informados ; campo e- mail omitido."	-Enviar um POST omitindo o campo emailVerificar resposta com erro.	Código: 400 Mensagem : "email é obrigatór io"	Resposta de erro com status 400 Mensagem de erro deve ser JSON Mensagem de erro correta	▼ PASSED
ID: US001- CT04	Tentativa de criação de usuário sem o campo PASSWORD Objetivo: Verificar se a API rejeita usuários sem senha.	Α	(MUST)	"API ativa; nome e e- mail válidos informados ; campo senha omitido."	Enviar um POST omitindo o campo passwordVerificar se retorna erro 400.	Código: 400 Mensagem : "password é obrigatór io"	✓ Resposta de erro com status 400 ✓ Resposta é um JSON válido ✓ Mensagem de erro correta para password	✓ PASSED
ID: US001- CT05	Tentativa de criação de usuário sem o campo ADMINISTR ADOR Objetivo: Garantir que o campo administrad or é obrigatório e validado.	A	(SHOULD)	"API ativa; nome, e- mail e senha válidos informados ; campo administra dor omitido."	Enviar um POST omitindo o campo administr ador Verificar se o sistema responde com erro.	Código: 400 Mensagem : "administ rador é obrigatór io"	Resposta de erro com status 400 Resposta é um JSON válido Mensagem de erro correta para administrador	✓ PASSED
ID: US001- CT06	Tentativa de criação de usuário com tipo inválido no campo ADMINISTR ADOR Objetivo: Verificar se o sistema valida os tipos de dados do campo	Α	(SHOULD)	"API ativa; nome, e- mail e senha válidos informados ; campo administra dor com valor inválido."	Enviar POST com administr ador: 123 (valor numérico inválido).	Código: 400 Mensagem : "administ rador deve ser 'true' ou 'false'"	Resposta de erro com status 400 Resposta é JSON válida Mensagem de erro correta para tipo inválido (número)	▼ PASSED

	administrad or.								
ID: US001- CT07	Login com usuário recém- cadastrado Objetivo: Verificar se o usuário criado com sucesso consegue autenticar com seu e- mail e senha.	A	(MUST)	Usuário válido cadastrad o.	Enviar uma requisição POST para /api/logi n com: email: "carlos@e mpresa.co m" password: "SenhaFort e123"	Código: 200 Corpo da resposta contém: "Login realizado com sucesso"	✓ Status da resposta é 200 ✓ Mensagem de login está correta	✓ PASSED	

Critério de aceite US002 🖉

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US002- CT01	Login com usuário inexistente Objetivo: Garantir que o sistema rejeita autenticação de um usuário que não está cadastrado.	Α	(MUST).	Pré- condição: Nenhum usuário com o e- mail testado existe no banco.	Enviar uma requisição POST para o endpoint /api/logi n com: email: naoexiste@ empresa.co m password: QualquerSe nha123 -Verificar se a resposta retorna 401 Unauthoriz ed -Verificar se a mensagem de erro informa	Código: 401 Mensagem: "Usuário e/ou senha inválidos"	✓ Login com usuário inexistente - Status 401 ✓ Mensagem de erro ao tentar login com usuário inexistente	▼ PASSED	

ID: US002- CT02	Consulta de usuário inexistente por ID Objetivo: Validar que não é possível consultar um usuário que não existe.	A	(SHOULD)	Pré- condição: O ID consultado não está no banco.	"Usuário e/ou senha inválidos" Enviar uma requisição GET para /api/usuar ios/9999 (ID inexistente) Verificar se a resposta retorna 404 Not Found	Código: 404 Mensagem: "Usuário não encontrado	X Status 404 para usuário inexistente AssertionError: expected response to have status code 404 but got 400 ✓ Mensagem correta para usuário inexistente	1 FALHOU X 1- PASSOU✓	Um erro encontrad o no processo de teste, quando se esperava um códico 404 é a api retornou um código
					Verificar se a mensagem informa "Usuário não encontrado				400, será aberta uma issue para as devidas investigaç ões é correções.
ID: US002- CT03	Atualização de usuário inexistente Objetivo: Garantir que o sistema não permite atualização de usuários que não existem.	A	(SHOULD)	Pré- condição: O ID informado na requisição não existe no banco.	Enviar uma requisição PUT para /api/usuar ios/9999 com dados válidos Verificar se retorna 404 Not Found Verificar se a mensagem informa "Usuário não encontrado "	Código: 404 Mensagem: "Usuário não encontrado "	★ Status 404 para usuário inexistente AssertionError: expected response to have status code 404 but got 201 ★ Mensagem correta para usuário inexistente AssertionError: expected 'Cadastro realizado com sucesso' to deeply equal 'Usuário não encontrado'	X 201 FALHOU	Um erro encontrad o no processo de teste, quando se esperava um códico 404 é a api retornou um código 201, será aberta uma issue para as devidas investigaç ões é correções.

ID: US002- CT04	Exclusão de usuário inexistente Objetivo: Verificar que não é possível deletar um usuário que não existe.	A	(SHOULD)	Pré- condição: O ID fornecido não está cadastrad o.	Enviar uma requisição DELETE para /api/usuar ios/9999 Verificar se retorna 404 Not Found Verificar se a mensagem informa "Usuário não encontrado "	Código: 404 Mensagem: "Usuário não encontrado "	X Status 400 para exclusão de usuário inexistente AssertionError: expected response to have status code 400 but got 200 X Mensagem correta para exclusão de usuário inexistente AssertionError: expected 'Nenhum registro excluído' to deeply equal 'Usuário não encontrado'	X 201 FALHOU	Apesar do status 201 o teste falhou, pois se esperava do sistema um codigo 404 de usuario não encontrad o. será aberto uma issue para investigaç ão e resolução do problema.
-----------------------	---	---	----------	---	--	--	---	--------------	--

Critério de aceite US003 🛭

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resulta do Espera do	Resultado Obtido	Status do teste	Observaç ões
ID: US003- CT01	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.	A	(MUST).	Pré- condições : Um usuário já existe com o e-mail joao@empr esa.com	Enviar uma requisição POST para /api/usuario s com os seguintes dados: nome: João da Silva email: joao@empresa .com password: SenhaSegura1 23 administrador : false	Código: 400 Mensag em: "Este e- mail já está sendo utilizado"	✓ Status 400 para e-mail já cadastrado ✓ Mensagem correta para e- mail já cadastrado	▼ PASSED	

					Verificar se a resposta retorna 400 Bad Request Verificar se a mensagem de erro informa que o e-mail já está em uso			
ID: US003- CT02	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 anterior com e-mail repetido.	A	(SHOULD)	Pré- condições : O e-mail joao@empr esa.com já está cadastrad o	Enviar uma requisição POST para /api/usuario s com: nome: João da Silva email: joao2@empres a.com password: SenhaSegura1 23 administrado r: false Verificar se a resposta retorna 201 Created Verificar se o usuário foi criado com um novo ID	Código: 201 Mensag em: Sucesso na criação do usuário	Revalidar cenario de teste, pois gerou uma incerteza de como o testador vai proceder nos testes.	

Critério de aceite US004 @

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US004- CT01	PUT com ID inexistente deve criar um novo usuario Objetivo:	A	(MUST)	O ID 9999 não existe no sistema	Enviar uma requisição PUT para /api/usuar	Código: 201 Mensagem: "Usuário	✓ Status code é 201 (Criado) ✓ Mensagem de sucesso está correta ✓ Corpo da resposta contém	✓ PASSED	

	Verificar que o sistema cria um novo usuário quando o ID passado no PUT não existe.				ios/9999 com: nome: Marina Souza email: marina@emp resa.com password: Senha123 administra dor: false Verificar se a resposta retorna 201 Created Verificar se um novo usuário foi criado com novo ID	criado com sucesso" Corpo da resposta deve conter os dados do novo usuário	o ID do novo usuário		
ID: US004- CT02	PUT com ID existente deve atualizar o usuário Objetivo: Garantir que o sistema atualiza o usuário se o ID informado já existir.	A	(MUST)	Pré- condições : Um usuário com ID 5 já está cadastrad o	Enviar uma requisição PUT para /api/usuar ios/5 com dados atualizados : nome: Marina Atualizad a email: marina@emp resa.com password: NovaSenha1 23 administra dor: true Verificar se a resposta	Código: 200 Mensagem: "Usuário atualizado com sucesso"		aguardand o os testes	

	retorna 200 OK	
	Verificar se os dados do usuário foram atualizados	

Critério de aceite US005 🛭

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US005- CT01	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.	A	(MUST)	Pré- condições : Usuário com ID 1 cadastrad o com e- mail joao@empr esa.com Usuário com ID 2 cadastrad o com e- mail maria@emp resa.com	Enviar uma requisição PUT para /api/usuar ios/2 com: nome: Maria Almeida email: joao@empre sa.com password: SenhaNova1 23 administra dor: 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 email já está sendo usado"	✓ Status code é 400 (Bad Request) ✓ Mensagem de erro está correta	▼ PASSED	
ID: US005-	PUT com o mesmo e-	A	(SHOULD)	Pré- condições	Enviar uma requisição	Código: 200	Status code é 200 (OK)	✓ PASSED	

os dados foram atualizados com sucesso
--

Critério de aceite US006 🔗

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US006- CT01	Registro de evidência para teste bem- sucedido Objetivo: Garantir que, ao executar um teste com sucesso, uma evidência (como log, print, ou	Α	(MUST)	Pré- condições : Ambiente de testes automatiz ado e configurad o Execução de testes via	Passos: Executar um teste positivo (ex: cadastro de usuário com dados válidos) Verificar se foi gerado arquivo de	Evidência salva automatica mente em diretório definido (ex: /evidencia s/testes_u suarios/) Nome do arquivo		aguardand o os testes	

	atório) seja rada.			ferramenta (ex: Postman, JMeter, Cypress, ou Jest)	evidência contendo: -Request enviada -Response recebida -Status code -Data/hora da execução -Resultado (Aprovado)	contendo o ID do teste e data		
US006- evi CT02 occ fall tes aut Obj Gai fall tes' tam ger evic	idência ao orrer uma ha em ste tomatizado jetivo: rantir que has em ttes	A	(MUST)	Pré- condições : Ambiente de testes automatiz ado configurad o para capturar falhas	Passos: Executa r um teste negativ o (ex: tentativ a de criar usuário com e- mail duplica do) Verificar se a evidênci a inclui: -Request enviada -Response recebida -Status code - Mensagem de erro -Stack trace (se aplicável)	Evidência armazenad a Nome do arquivo indicando que foi uma falha.	aguardand o os testes	
US006- sal	idência Iva com nt de tela	A	(SHOULD)	Pré- condições :	Passos: • Realizar o teste	Print salvo Associado corretamen	aguardand o os testes	

(para	testes	Testes	manual	te no
manu		sendo	mente	relatório/m
Objet	ivo:	executado	(ex: via	anual de
Garar	ntir que	S	frontend	testes
testes	5	manualme)	
manu	ıais	nte	• Captura	
possu	uam		r print	
scree	nshots		de tela	
como	·		da	
evidê	ncia.		operaçã	
			o final	
			• Salvar o	
			print	
			com o	
			ID do	
			caso de	
			teste	
			• Associa	
			r o print	
			ao caso	
			de teste	
			no	
			plano	
			de	
			testes	

Critério de aceite US007

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US007- CT01	Tentativa de cadastro com e-mail do Glmai Objetivo: Verificar se o sistema bloqueia o cadastro de usuários com e-mail do provedor Gmail.	Α	(MUST)	Pré- condições : API de cadastro de usuário disponível.	Passos: Enviar uma requisição POST para /api/usuar ios com os seguintes dados: nome: "Ana Silva" email: "ana@gmail.com"	Código: 400 Mensagem: "Não é permitido o uso de e- mails do provedor gmail."	X Status code é 400 (Bad Request) AssertionError: expected response to have status code 400 but got 201 X Mensagem de erro está correta AssertionError: expected 'Cadastro realizado com sucesso' to deeply equal 'Não é permitido o uso de e-mails do p'	X FAIL	A API permitiu a criação com @gmail o que vai em desacordo com a proposta na document ação, será aberto uma issue para dar

					password: "Senha123" administra dor: "false" Verificar a resposta da API.				inicio as investigaç ões e possiveis correções.
ID: US007- CT02	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.	A	(MUST)	Pré- condições : API de cadastro de usuário disponível.	Enviar uma requisição POST para /api/usuar ios com os seguintes dados: nome: "Bruno Costa" email: "bruno@ho tmail.com" password: "Senha123" administra dor: "true" Verificar a resposta da API.	Código: 400 Mensagem: "Não é permitido o uso de e- mails do provedor hotmail."	X Status code é 400 (Bad Request) AssertionError: expected response to have status code 400 but got 201 X Mensagem de erro está correta AssertionError: expected 'Cadastro realizado com sucesso' to deeply equal 'Não é permitido o uso de e-mails do p'	X FAIL	A API permitiu a criação com @hotmail o que vai em desacordo com a proposta na document ação, será aberto uma issue para dar inicio as investigaç ões e possiveis correções.
ID: US007- CT03	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.	A	(MUST)		Passos: Enviar um POST para /api/usuar ios com e-mail "joao@emp resa.com". Verificar a resposta.	Código: 201 Usuário criado com sucesso.	✓ Status code é 201 (Criado) ✓ Mensagem de sucesso está correta ✓ Corpo da resposta contém o ID do usuário	▼ PASSED	

Critério de aceite US008 @

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US008- CT01	Cadastro com e-mail válido Objetivo: Garantir que o sistema aceite e-mails válidos no formato correto.	A	(MUST)	Pré- condições : API de cadastro disponível.	Passos: Enviar uma requisição POST para /api/usuar ios com: nome: "Lucas Lima" email: "lucas@em presa.com" password: "SenhaSeg ura123" administra dor: "true" Verificar a resposta.	Código: 201 Usuário cadastrado com sucesso.	✓ Status code é 201 (Criado) ✓ Usuário cadastrado com sucesso ✓ Resposta contém ID do usuário	✓ PASSED	
ID: US008- CT02	Cadastro com e-mail sem o caractere @ Objetivo: Verificar se o sistema rejeita e-mails sem o caractere "@".	Α	(MUST)	Pré- condições : API de cadastro disponível.	Passos: Enviar uma requisição POST para /api/usuar ios com o e-mail: "joaoempre sa.com". Verificar a resposta.	Código: 400 Mensagem: "Formato de e-mail inválido."	✓ Status 400 - E-mail inválido ✓ Mensagem de erro correta	✓ PASSED	
ID: US008- CT03	Cadastro com e-mail sem domínio Objetivo: Verificar se o sistema rejeita e-mails sem domínio.	Α	(MUST)	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."	✓ Status 400 - E-mail sem domínio ✓ Mensagem contém 'Formato de e-mail inválido' AssertionError: expected 'email é obrigatório' to include 'inválido'	1-PASSED X	Apesar da API ter retornado o status code 400 mensagen s de erro foram disparada s, fazendo

									assim não completar os teste.
ID: US008- CT04	Cadastro com e-mail sem nome de usuário Objetivo: Validar a rejeição de e- mails sem nome de usuário antes do @.	A	(MUST)	Pré- condições : API de cadastro disponível.	Passos: Enviar um POST com o e-mail: "@empres a.com". Verificar a resposta.	Código: 400 "email deve ser um email válido"	✓ Status 400 - E-mail sem nome de usuário ✓ Mensagem de erro correta para e-mail inválido	✓ PASSED	

Critério de aceite US009

ID	Casos de Teste	Tip o M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US009- CT01	Cadastro com senha dentro do limite de caracteres (5 caracteres) Objetivo: Verificar se o sistema aceita senhas com 5 caracteres.	Α	(MUST)	Pré- condições : API de cadastro disponível.	Passos: Enviar uma requisição POST para /api/usuar ios com os seguintes dados: nome: "Carlos Souza" email: "carlos@e mpresa.co m" password: "12345" administra dor: "false" Verificar a resposta.	Código: 201 Usuário cadastrado com sucesso.	✓ Status 201 - Cadastro com senha de 5 caracteres ✓ Usuário cadastrado com sucesso ✓ Verificação do tamanho da senha	▼ PASSED	
ID: US009-	Cadastro com senha	A	(MUST)	Pré- condições	Enviar uma requisição	Código: 400	X Status 400 - Senha com menos de 5	1- PASSED 2- FAIL X	O sistema permitiu a

CT02	abaixo do limite de caracteres (menos de 5 caracteres) Objetivo: Verificar se o sistema rejeita senhas com menos de 5 caracteres.			: API de cadastro disponível.	POST para /api/usuar ios com os seguintes dados: nome: "Renato Lima" email: "renato@e mpresa.co m" password: "123" administra dor: "true" Verificar a resposta.	Mensagem: "A senha deve ter entre 5 e 10 caracteres. "	caracteres AssertionError: expected response to have status code 400 but got 201 X Mensagem de erro da senha AssertionError: expected 'Cadastro realizado com sucesso' to include 'senha deve ter entre 5 e 10 caracteres' Y Confirma tamanho inválido da senha		criação se senha com menos de cinco caracteres , por isso será aberto uma issue para verificar o possivel erro, assim podendo tamém ser executada todas as correções necessari as.
ID: US009- CT03	Cadastro com senha no limite máximo de caracteres (10 caracteres) Objetivo: Verificar se o sistema aceita senhas com 10 caracteres.	A	(MUST)	Pré- condições : API de cadastro disponível.	Passos: Enviar uma requisição POST para /api/usuar ios com os seguintes dados: nome: "Felipe Almeida" email: "felipe@em presa.com" password: "senha123 45" administra dor: "false" Verificar a resposta.	Código: 201 Usuário cadastrado com sucesso.	✓ Status 201 - Cadastro com senha de 10 caracteres ✓ Usuário cadastrado com sucesso ✓ Verificação do tamanho da senha	PASSED V	
ID: US009- CT04	Cadastro com senha acima do limite de caracteres	A	(MUST)	Pré- condições :	Passos: Enviar uma requisição POST para	Código: 400 Mensagem: "A senha	Senha excede limite de caracteres AssertionError: expected response to have	1- PASSED 2- FAIL X	O sistema permitiu a criação se senha com mais

(mais de 10	API de	/api/usuar	deve ter	status code 400 but got 201	de dez	
caracteres)	cadastro	ios com	entre 5 e 10	★ Mensagem	caracteres	
Objetivo:	disponível.	os	caracteres.	de validação da	, por isso	
Verificar se o		seguintes	п	senha	será	
sistema		dados:		AssertionError: expected	aberto	
rejeita senhas				'Cadastro	uma issue	
com mais de		nome:		realizado com sucesso' to	para	
10 caracteres.		"Juliana		include 'senha	verificar o	
10 001 00100		Costa"		deve ter entre 5	possivel	
		email:		e 10 caracteres'	'	
				Confirma tamanho inválido	erro,	
		"j <u>uliana@e</u>		da senha	assim	
		mpresa.co			podendo	
		<u>m</u> "			tamém ser	
		password:			executada	
		"senha123			todas as	
		45678"			correções	
		43076			necessari	
		administra			as.	
		dor: "true"			40.	
		Verificar a				
		resposta.				

Critério de aceite US010 🔗

ID	Casos de Teste	Ti po M/ A	Prioridad e	Pré- condições	Passos	Resultado Esperado	Resultado Obtido	Status do teste	Observaç ões
ID: US010- CT01	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.	A	(MUST)	Pré- condições : -API de cadastro disponívelSwagger da API acessível.	Testar o endpoint POST /api/usuar ios enviando dados inválidos para os campos obrigatório s. Exemplo: Enviar um e-mail inválido, senha muito curta, ou valores	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).			Ao decorrer do ciclo de testes, testes que foram realizados cobrem bem esse cenário, criterio de aceite deve ser reavaliado .

					inválidos para os campos. Testar o endpoint PUT /api/usuar ios/{id} com ID inválido ou com dados inconsisten tes. Testar o endpoint DELETE /api/usuar ios/{id} tentando deletar um usuário inexistente.	Mensagens de erro claras e apropriada s, como "E- mail inválido", "ID não encontrado ", etc.	
ID: US010- CT02	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.	A	(MUST)	Pré- condições : API de login e cadastro de usuários disponívei s. Swagger da API acessível.	Realizar um login com um usuário válido e verificar se o token é retornado corretamen te. Tentar acessar um endpoint restrito (por exemplo, listagem de usuários) sem autenticaç ão e verificar se a resposta é 401 (Unauthori zed). Tentar acessar o mesmo	A resposta para um login válido deve ser 200 (OK) com o token Bearer. A resposta para um acesso sem autenticaç ão deve ser 401 (Unauthori zed). A resposta para um acesso com token válido deve ser 200 (OK)	Ao decorrer do ciclo de testes, testes que foram realizados cobrem bem esse cenário, criterio de aceite deve ser reavaliado .

					endpoint com um token válido e verificar se o acesso é permitido.			
ID: US010- CT03	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.	A	MUST)	Pré- condições : API de cadastro disponível. Swagger da API acessível.	Enviar um grande volume de requisições POST para /api/usuar ios com dados válidos e validar se o sistema consegue processálas sem erros. Enviar uma requisição POST para /api/usuar ios com dados válidos repetidame nte (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 corretamen te 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 apropriada s quando os limites de dados forem excedidos.	aguardand o os testes	