

# Plano de testes

## Sistema de envio de feedbacks

### 1. Introdução

Esse sistema visa permitir que colaboradores de uma empresa possam se cadastrar e enviar feedbacks uns para os outros, apontando pontos positivos e de melhoria em relação ao trabalho. Cada usuário possui seu perfil, onde ficam os feedbacks recebidos e enviados. É possível visitar a página de perfil de outros colaboradores e visualizar os seus feedbacks.

Existe a possibilidade de enviar um feedback anônimo, em que o nome de quem enviou não é divulgado. Também é possível escolher se um feedback recebido ficará visível ou não em seu perfil.

### 2. Funcionalidades para teste

Identificamos as principais funcionalidades do sistema como sendo: cadastro de usuário, login e criação de feedbacks. Dessa forma, foram elaborados testes de tela para garantir que essas funções estejam de acordo com o que foi acordado na documentação do projeto.

Também criamos testes para todos os endpoints criados pelo time de backend. Com esses testes garantimos a qualidade no envio de requests e nas informações que recebemos do banco de dados.

### 3. Escopo

Realizaremos testes de funcionalidade nas seguintes interfaces do sistema:

- Tela de cadastro do usuário
- Tela de login do usuário
- Tela de criação de feedback
- Tela de perfil do usuário

Faremos também testes para todos os endpoints que pertencem às controllers abaixo:

- users-controller
- feedback-controller
- tags-controller

## 4. Estratégia de teste

Realizaremos testes de função nas interfaces e testes de API nos endpoints das controllers apontadas acima. O teste de função tem como objetivo garantir que as regras de negócio estão sendo aplicadas de maneira apropriada em cada funcionalidade do sistema. Os testes na API garantem seu funcionamento de maneira correta. Quando formos testar as telas, usaremos técnicas de caixa preta, onde iremos interagir com o aplicativo por meio de interface gráfica.

## 5. Ferramentas

Usamos como ferramentas o Rest-assured para testar a API e o Selenium para testes de interface. Além disso, usamos o Trello para organização da equipe e apontamento de bugs do sistema.

## 6. Cenários testados

### Teste de API:

✓ UserAceitacao	29 sec 935 ms
✓ create	7 sec 325 ms
✓ createEmailExiste	2 sec 356 ms
✓ createSemCargo	730 ms
✓ createSemEmail	767 ms
✓ createSemEmailDBC	754 ms
✓ createSemNome	750 ms
✓ createSemSenha	745 ms
✓ deleteUser	1 sec 571 ms
✓ fileUpdate	2 sec 501 ms
✓ fileUpdateEmpty	2 sec 350 ms
✓ listLoggedUser	742 ms
✓ userListAll	740 ms
✓ userListById	2 sec 309 ms
✓ userLogin	3 sec 113 ms
✓ userLoginEmailIncorreto	811 ms
✓ userLoginSenhaIncorreto	2 sec 371 ms

✓ FeedbackAceitacao	14 sec 619 ms
✓ createFeedbackComSucesso	4 sec 372 ms
✓ createFeedbackSemMensagem	869 ms
✓ createFeedbackSemStatus	882 ms
✓ createFeedbackSemTagComSucesso	1 sec 290 ms
✓ createFeedbackSemUser	831 ms
✓ createFeedbackUsuarioInexistente	811 ms
✓ givenFeedbacks	1 sec 451 ms
✓ givenFeedbacksById	865 ms
✓ receivedFeedbacks	838 ms
✓ receivedFeedbacksById	809 ms
✓ updateVisibility	778 ms
✓ updateVisibilityFeedbackInexistente	823 ms

✓ TagAceitacao	34 sec 99 ms
✓ listTags	34 sec 99 ms

## Testes de telas:

✓	✓	CadastroSteps (steps)	1 min 25 sec
	✓	cadastrarComSucesso	16 sec 971 ms
	✓	cadastrarSemConfirmacaoSenha	7 sec 455 ms
	✓	cadastrarSemImagemComSucesso	13 sec 735 ms
	✓	cadastrarSemNome	6 sec 971 ms
	✓	cadastrarSemEmail	10 sec 535 ms
	✓	cadastrarSemSenha	12 sec 558 ms
	✓	cadastrarSenhasDiferentes	16 sec 630 ms
✓	✓	EnviarFeedbackSteps (steps)	2 min 37 sec
	✓	criarFeedbackSemDestinatario	16 sec 106 ms
	✓	criarFeedbackSemTagComSucesso	19 sec 146 ms
	✓	criarFeedbackAnonimoComSucesso	27 sec 597 ms
	✓	criarFeedbackComSucesso	25 sec 869 ms
	✓	criarFeedbackSemMensagem	50 sec 898 ms
	✓	criarFeedbackUsuarioInexistente	17 sec 268 ms
✓	✓	LoginSteps (steps)	1 min 39 sec
	✓	loginSemEmail	13 sec 117 ms
	✓	loginSemSenha	12 sec 281 ms
	✓	loginComEmailErrado	30 sec 901 ms
	✓	loginSenhaErrada	15 sec 266 ms
	✓	loginComSucesso	27 sec 156 ms