Referência: avaliaser-qa/AvaliaSer - Plano de Teste.docx.pdf at main · vemser/avaliaser-qa (github.com)

### 1. Introdução

1.1. Este documento possui o plano de teste elaborado para o AvaliaSer, o qual é um sistema de avaliação dos alunos afim de auxiliar a equipe de gestão de pessoas e instrutores do Vem Ser.

### 2. Arquitetura

- 2.1. O framework utilizado para a implementação do sistema AvaliaSer é o React JS, que possui uma arquitetura orientada a componentes. Para o armazenamento, consulta e alteração de dados da aplicação, será usada uma API que disponibiliza um conjunto de serviços REST. Em que o cliente (aplicação) envia uma requisição à uma URL (endpoint) para acessar os serviços da API, e tem como retorno um objeto na forma de um JSON.
- 2.2. Referências/Documentações que farão parte do Projeto:
  - 2.2.1. Especificação de Requisitos
    - 2.2.1.1. Irá definir as funcionalidades e o ambiente que o Sistema deve possuir
  - 2.2.2. Especificação de Casos de Uso
    - 2.2.2.1. Descreve de forma detalhada as funcionalidades do Sistema
  - 2.2.3. Especificação das Regras de negócio
    - 2.2.3.1. Definirá as regras e os comportamentos do Sistema

### 3. Requisitos

### 3.1. Requisitos Funcionais

#### 3.1.1. Usuários

- → O sistema deve permitir que o usuário faça login no sistema via ECOS.
- → O sistema deve mostrar o respectivo ambiente de cada classe de usuário.
  - 3.1.2. Estagiários
- → O sistema deve permitir que o estagiário visualize os acompanhamentos disponíveis.
- → O sistema deve permitir que o estagiário agende um acompanhamento disponível.
- → O sistema deve permitir que o estagiário visualize um acompanhamento agendado.
- → O sistema deve permitir que o estagiário visualize seu histórico de acompanhamentos.
  - 3.1.3. GPs
- → O sistema deve permitir que o GP tenha acesso ao CRUD dos estagiários.
- → O sistema deve permitir que o GP importe os estagiários do setor de Captação.
- → O sistema deve permitir que o GP tenha acesso ao CRUD dos acompanhamentos.
- → O sistema deve permitir que o GP tenha um CRUD dos agendamentos dos estagiários.
  - 3.1.4. Instrutores
- → O sistema deve permitir que o Instrutor visualize os acompanhamentos.
- → O sistema deve permitir que o Instrutor tenha acesso ao CRUD de feedbacks técnicos.
  - 3.1.5. Administrador
- → O sistema deve permitir que o Administrador tenha acesso a todos os recursos dos GPs e Instrutores.
  - 3.2. Requisitos Não Funcionais

### 4. Funcionalidades

4.1. Página Inicial Avaliação (Acessando como Admin, GP ou Instrutor)

- → Deve ter um cabeçalho com a foto e o nome do usuário, seguindo o padrão de mensagem: "Olá, <nome do usuário>".
- → Menu com as seguintes opções:
  - Dashboard
  - Comportamental
  - Técnico
  - o Configurações
  - Voltar
  - 4.2. Página Inicial Avaliação (Acessando como Estagiário)
- → Deve ter um cabeçalho com a foto e o nome do usuário, seguindo o padrão de mensagem: "Olá, <nome do usuário>".
- → Menu com as seguintes opções:
  - Dashboard
  - Meus Acompanhamentos
  - Minhas Tasks
  - o Voltar

# 5. Estratégias de Teste

- 5.1. Escopo de Testes
  - 5.1.1. O plano de testes abrange todas as funcionalidades descritas na tabela acima. Serão executados testes em todos os níveis conforme a descrição abaixo.
- 5.2. Testes Unitários: -
  - 5.2.1. O backend terá uma cobertura de 100% de testes unitários na camada de serviço (service), que são de responsabilidade dos desenvolvedores backend. -
  - 5.2.2. O frontend terá uma cobertura de 60% de testes unitários, que são de responsabilidade dos desenvolvedores frontend.
- 5.3. Testes de Integração: serão executados testes de integração em todos os endpoints, e esses testes serão de responsabilidade do time de qualidade.
- 5.4. Testes Manuais: funcionalidades serão testadas manualmente pelo time de qualidade seguindo o Cenário de Testes deste plano de testes. Serão feitos testes manuais nas funcionalidades: Recuperar senha do usuário; Mostrar senha; Cadastrar Feedback; Cadastrar Acompanhamento e Avaliar Acompanhamento.
- 5.5. Testes Automatizados: serão realizados testes end-to-end em todas as funcionalidades, seguindo o Cenário de Testes deste plano de testes, sob responsabilidade do time de qualidade

### 6. Ambiente e Ferramentas

- 6.1. Testes Automatizados
  - 6.1.1. Gerenciamento
    - 6.1.1.1. Jenkins Allure Reports
  - 6.1.2. Back-end
    - 6.1.2.1. RestAssured Java
  - 6.1.3. Front-end
    - 6.1.3.1. Selenium 4.8.3 RestAssured Java 17
    - 6.1.3.2. ChromeDriver 111
- 6.2. Testes Exploratórios/Manuais
  - 6.2.1. Postman
  - 6.2.2. Swagger

### 7. Classificação de Bugs

- 7.1. Crítico
  - Bug que bloqueia o teste de uma função ou feature causa crash na 7.1.1. aplicação
  - 7.1.2. Botão não funciona impedindo o uso completo da funcionalidade
  - 7.1.3. Bloqueia a entrega
- 7.2. Grave
  - 7.2.1. Funcionalidade não funciona como o esperado
  - 7.2.2. Input incomum causa efeitos irreversíveis
- 7.3. Moderada
  - Funcionalidade não atinge certos critérios de aceitação, mas sua 7.3.1. funcionalidade em geral não é afetada
  - Mensagem de erro ou sucesso não é exibida 7.3.2.
  - 7.3.3.
- 7.4. Pequena
  - 7.4.1. Quase nenhum impacto na funcionalidade, porém atrapalha a experiência
  - 7.4.2. Erro ortográfico
  - Pequenos erros de UI 7.4.3.
  - 7.4.4. Status code dentro do intervalo esperado, mas não corresponde ao padrão ideal.

### 8. Definição de Pronto

8.1. Será considerada pronta as funcionalidades que passarem pelas verificações e testes descritos neste plano de teste, não apresentarem bugs com a severidade acima de "moderada", e passarem por uma validação de negócio de responsabilidade do time de produto.

### 9. Cronograma

- 9.1. Etapa 1
  - 9.1.1. Descrição
    - Validação de Diagrama ER e Figma 9.1.1.1.
    - Elaboração de plano de teste 9.1.1.2.
  - 9.1.2. Prazo
    - 9.1.2.1. 06/04/23 > 09/04/23
- 9.2. Etapa 2
  - 9.2.1. Descrição
    - 9.2.1.1. Preparação de ambientes de teste (Jenkins, Selenium, RestAssured, Allure)
  - 9.2.2. Prazo
    - 9.2.2.1. 06/04/23 > 10/04/23
- 10. Cenários de Testes
  - 10.1. Testes voltados para API

#### ET01 - Estagiário CRUD Básico

ET01 - CT01 - Criar estagiário com sucesso 🔔 🚾



Dado que o path do endpoint está correto

E usuário está autenticado

Quando definir body JSON com chaves válidas do estagiário

Então a resposta deve conter status code 201 (ACEITÁVEL 200)

E body em JSON com atributos do estagiário.

### ET01 - CT02 - Criar estagiário com CPF com ponto e hífen 🔔 🔯

Dado que o path do endpoint está correto

E usuário está autenticado

Quando definir body JSON com chave "cpf" inválido ("071.545.751-90")

E demais chaves válidas do estagiário

E requisitar POST

Então a resposta deve conter status code 400

E body em JSON com atributos com timestamp, status e erros respectivos.

### ET01 - CT03 - Criar Estagiário com Email Pessoal inválido 🗘 🗽





Dado que o path do endpoint está correto

E usuário está autenticado

Quando definir body JSON com chave "emailPessoal" inválido

E demais chaves válidas do estagiário

E requisitar POST

Então a resposta deve conter status code 400

E body em JSON com atributos com timestamp, status e erros respectivos.

### ET01 - CT04 - Criar Estagiário com Email Corporativo inválido 🔔 🚾





Dado que o path do endpoint está correto

E usuário está autenticado

Quando definir body JSON com chave "emailCorporativo" inválido

E demais chaves válidas do estagiário

E requisitar POST

Então a resposta deve conter status code 400

E body em JSON com atributos com timestamp, status e erros respectivos.

### ET01 - CT05 - Buscar estagiário por ID com sucesso



Dado que o endpoint é "/estagiario/get-by-id/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

Quando requisitar GET

Então a resposta deve conter status code 200

E body em JSON com atributos do estagiário.

ET01 - CT06 - Buscar estagiário por ID inexistente **Dado** que o endpoint é "/estagiario/get-by-id/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

Quando requisitar GET

Então a resposta deve conter status code 404.

### ET01 - CT07 - Buscar estagiário por ID inválido



**Dado** que o endpoint é "/estagiario/get-by-id/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inválido (diferente de número maior que 0) Quando requisitar GET

Então a resposta deve conter status code 400.

### ET01 - CT08 - Buscar estagiário por programa

**Dado** que o endpoint é "/estagiario/list-by-programa/{idPrograma}"

E usuário está autenticado

E o path contém "idPrograma" existente

E o query contém "pagina" com número maior ou igual a zero

E o query contém "tamanho" com número maior ou igual a um

Quando requisitar GET

Então a resposta deve conter status code 200

E body em JSON com número de páginas informado

E tamanho informado

E estagiários pertencentes ao programa nesta página e tamanho.

ET01 - CT09 - Buscar estagiário por programa com página e tamanho inválidos 🔔 🔯

**Dado** que o endpoint é "/estagiario/list-by-programa/{idPrograma}"

E usuário está autenticado

E o path contém "idPrograma" existente

E o query contém "pagina" com valor inválido

E o query contém "tamanho" com valor inválido

Quando requisitar GET

Então a resposta deve conter status code 400.

## ET01 - CT10 - Buscar por lista de todos estagiários or

**Dado** que o endpoint é "/estagiario/list-all"

E usuário está autenticado

E o query contém "pagina" com número maior ou igual a zero

E o query contém "tamanho" com número maior ou igual a um

Quando requisitar GET

Então a resposta deve conter status code 200

E body em JSON com número de páginas informado

E tamanho informado

E estagiários nesta página e tamanho.

### ET01 - CT11 - Buscar por lista de todos estagiários com página e tamanho inválidos

Dado que o endpoint é "/estagiario/list-all"

E usuário está autenticado

E o query contém "pagina" com valor inválidos

E o query contém "tamanho" com valor inválidos

Quando requisitar GET

Então a resposta deve conter status code 400.

### ET01 - CT12 - Atualizar estagiário com sucesso

**Dado** que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado

Quando definir body JSON com chaves válidas do estagiário

E requisitar PUT

Então a resposta deve conter status code 200

E body em JSON com informações atualizadas do estagiário.

### ET01 - CT13 - Atualizar estagiário inexistente

**Dado** que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

E o body contém informações válidas do estagiário a ser atualizado

Quando definir body JSON com chaves válidas do estagiário

E requisitar PUT

Então a resposta deve conter status code 404

E mensagem de erro informando que o estagiário não foi encontrado.

### ET01 - CT14 - Atualizar estagiário com Email Pessoal inválido

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado, exceto pelo email pessoal inválido

Quando requisitar PUT

Então a resposta deve conter status code 400

E mensagem de erro informando que o email pessoal é inválido.

### ET01 - CT15 - Atualizar estagiário com Email Corporativo inválido 🚾

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado, exceto pelo email corporativo inválido

Quando requisitar PUT

Então a resposta deve conter status code 400

E mensagem de erro informando que o email corporativo é inválido.

### ET01 - CT16 - Atualizar estagiário com CPF inválido

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado, exceto pelo CPF inválido

Quando requisitar PUT

Então a resposta deve conter status code 400

E mensagem de erro informando que o CPF é inválido.

### ET01 - CT17 - Deletar estagiário com sucesso 🔔 🔯



Dado que o endpoint é "/estagiario/delete/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

Quando requisitar DELETE

Então a resposta deve conter status code 204 (ACEITÁVEL 200).

### ET01 - CT18 - Deletar estagiário inexistente



Dado que o endpoint é "/estagiario/delete/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 404.

### ET01 - CT19 - Desativar estagiário



Dado que o endpoint é "/estagiario/deactivate/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

Quando requisitar DELETE

Então a resposta deve conter status code 200

E body JSON com dados do usuário com atributo "ativo" igual a "false".

### ET01 - CT20 - Desativar estagiário já desativado ok



Dado que o endpoint é "/estagiario/deactivate/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o estagiário já está desativado

#### Quando requisitar DELETE

Então a resposta deve conter status code 404

ET01 - CT21 - Desativar estagiário por ID inexistente

Dado que o endpoint é "/estagiario/deactivate/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 404

ET01 - CT22 - Atualizar estagiário com Nome inválido 🚣

Cenário: Atualizar estagiário com Nome inválido

Dado que o path do endpoint para atualização de estagiários está definido

E existe um estagiário cadastrado com ID válido

E o body da requisição contém um campo "nome" inválido

Quando enviar uma requisição PUT com os dados do estagiário atualizados

Então a resposta deve conter status code 400

E o body em JSON deve conter os atributos "timestamp", "status" e "erros" com as informações correspondentes ao erro de validação do campo "nome".

### ET02 - Programa CRUD Básico

ET02 - CT01: Criar programa com sucesso

Dado que o endpoint é "/programa/create"

E usuário está autenticado como administrador ou GP

E o body contém informações válidas do programa a ser criado

Quando requisitar POST

Então a resposta deve conter status code 201

E body em JSON com informações do programa criado.

ET02 - CT02 - Falha ao criar um programa com nome nulo



Dado que o endpoint é "/programa/create"

E usuário está autenticado como administrador ou GP

E o body contém informações válidas do programa a ser criado, exceto pelo nome nulo

Quando requisitar POST

Então a resposta deve conter status code 400

E mensagem de erro informando que o nome é obrigatório

ET02 - CT03 - Falha ao criar um programa com datas abaixo da atual



Dado que o endpoint é "/programa/create"

E usuário está autenticado como administrador ou GP

E o body contém informações válidas do programa a ser criado, exceto pela data de início e data de fim abaixo da data atual

Quando requisitar POST

Então a resposta deve conter status code 400

E mensagem de erro informando que as datas de início e fim devem ser posteriores à data atual.

ET02 - CT04: Editar programa com sucesso



Dado que o endpoint é "/programa/update/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" existente

E o body JSON contém novas informações válidas do programa

Quando requisitar PUT

Então a resposta deve conter status code 200

E body em JSON com informações atualizadas do programa.

ET02 - CT05 - Falha ao editar programa com id vazio 🔔 🚾

Dado que o endpoint é "/programa/update/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" vazio

E o body JSON contém novas informações válidas do programa

Quando requisitar PUT

Então a resposta deve conter status code 400 (ACEITAVEL 404)

ET02 - CT06 - Falha ao editar programa com id inexistente

Dado que o endpoint é "/programa/update/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inexistente

E o body JSON contém novas informações válidas do programa

Quando requisitar PUT

Então a resposta deve conter status code 404

E mensagem de erro informando que o programa não foi encontrado.

ET02 - CT07: Desativar programa com sucesso

Dado que o endpoint é "/programa/deactivate/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" existente

Quando requisitar DELETE

Então a resposta deve conter status code 201

ET02 - CT08 - Falha ao desativar programa com letras no id Dado que o endpoint é "/programa/deactivate/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inválido com letras

Quando requisitar DELETE

Então a resposta deve conter status code 404

ET02 - CT09 - Falha ao desativar programa com id inexistente

Dado que o endpoint é "/programa/deactivate/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 400 (ACEITÁVEL 404)

ET02 - CT10: Deletar programa com sucesso



Dado que o endpoint é "/programa/delete/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" existente

Quando requisitar DELETE

Então a resposta deve conter status code 204

ET02 - CT11: Falha ao deletar programa com letras no id

Dado que o endpoint é "/programa/delete/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" com letras

Quando requisitar DELETE

Então a resposta deve conter status code 400

E mensagem de erro informando que o id deve ser numérico.

ET02 - CT12: Falha ao deletar programa com id inexistente Dado que o endpoint é "/programa/delete/{idPrograma}"



E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 404

E mensagem de erro informando que o programa não foi encontrado.

#### Testes automatizados voltados para WEB

### FEATURE: LOGIN

Usuário faz login - CT01 ok

Cenário: Usuário faz login com sucesso Dado que o usuário está na página de login

Quando o usuário preencher o campo de usuário com um usuário válido

E o usuário preencher o campo de senha com uma senha válida

E o usuário clicar no botão de login

Então o usuário deve ser redirecionado para a página de acesso

E o usuário deve estar logado como o usuário válido

E o nome do usuário logado deve ser exibido como o nome do usuário válido

Usuário faz login - CT02 OK



Cenário: Usuário faz logout de sua conta com sucesso

Dado que o usuário está logado em sua conta

Quando o usuário clicar no ícone de usuário

E o usuário clicar no botão de logout

Então o usuário deve ser redirecionado para a página de login

E a página de login deve ser exibida

Usuário faz login - CT03 OK



Cenário: Usuário faz login com dados inválidos

Dado que o usuário está na página de login

Quando o usuário preencher os campos de usuário e senha com usuário válido e senha inválida

Ou usuário inválido e senha válida Ou usuário inválido e senha inválida

E o usuário clicar no botão de login

Então o usuário deve permanecer na página de login E uma mensagem de sessão expirada deve ser exibida

### FEATURE: NAVEGAÇÃO PELO MENU DE AVALIAÇÃO (ADM)

### Navegar pelo menu como Administrador – CT01

Cenário: Acessar Dashboard via menu

Dado que o usuário está logado

E que está na plataforma de Avaliação

Quando clicar no ícone do menu

E selecionar a opção "Dashboard"

Então deve ser redirecionado para a página de Dashboard

E deve ser exibida a informação "Dashboard" no topo da página

Cenário: Acessar Estagiários via menu

Dado que o usuário está logado

E que está na plataforma de Avaliação

Quando clicar no ícone do menu

E selecionar a opção "Estagiários"

Então deve ser redirecionado para a página de Estagiários

E deve ser exibida a informação "Estagiários" no topo da página

### FEATURE: INTEGRAÇÃO COM API CAPTAÇÃO

Integração API Captação - CT01 :warning:

Cenário: Preencher cadastro de um estagiário a partir de um CPF de candidato da Captação

Dado que o usuário está na página de estagiários

Quando o usuário clicar no botão de cadastro de estagiário

E o usuário preencher o campo de busca de CPF com um CPF de candidato da Captação do Programa Atual

E o usuário clicar no botão de busca de CPF

Então os dados do candidato devem ser exibidos nos devidos campos em comum com estagiário

E os demais campos do estagiário devem ser visíveis para preenchimento

Integração API Captação - CT02 OK

Cenário: Cadastro de estagiário por busca de CPF não cadastrado na Edição Atual

Dado que o usuário está na página de cadastro de estagiários

Quando o usuário preencher o campo de busca de CPF com um CPF não cadastrado na Edição Atual

E o usuário clicar no botão de busca de CPF

Então o sistema deve exibir uma mensagem informando que o CPF não consta nos registros da Captação.

FEATURE: Dashboard (Administrador, GP)

Ranking de estagiários na página Dashboard – CT01

Cenário: Validar ranking de estagiários na página Dashboard

Dado que o usuário está logado

E que está na plataforma de Avaliação

Quando acessar a página de Dashboard

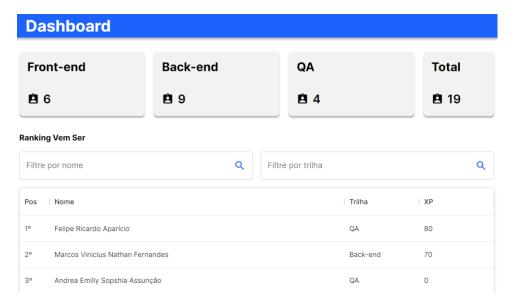
Então deve ser exibido o ranking de estagiários, ordenado pelo desempenho na avaliação

E cada estagiário deve ser exibido com seu nome e posição no ranking.

# 11. Relatórios de Testes Exploratórios

• Ranking de estagiários na página Dashboard – Dashboard – CT01

Estagiários são listados automaticamente por seu XP ao acessar página de Dashboard como Administrador.



Ao alterar critério de ordem para alfabético e/ou buscar estagiários por filtro de nome, a posição no rankink é conservada conforme esperado.



• Registro de feedback do GP – CT01 - :ok:

Cenário: Registrar um feedback do GP com sucesso

Dado que o usuário está logado como GP ou Administrador

E que está na página de Avaliação

Quando acessar a página Comportamental

E acessar a página Acompanhamentos

E clicar em "Avaliar" ao lado do estagiário para o qual deseja registrar o feedback

E preencher o formulário com as informações necessárias

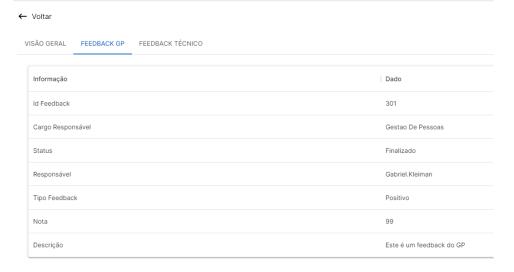
E clicar em "Cadastrar"

Então deve ser exibida uma mensagem de confirmação de que o feedback foi registrado com sucesso

E o feedback deve ser registrado no sistema, associado ao estagiário selecionado.



### Informações Avaliação



- Consulta de feedbacks pelos GP CT02 :ok:
- Registro de feedback Técnico-CT01 :ok:

Cenário: Registrar um feedback técnico com sucesso

Dado que o usuário está logado como Instrutor ou Administrador

E que está na página de Avaliação

Quando acessar a página "Técnico"

E acessar a página "Feedback"

E acessar um Acompanhamento válido

E clicar em "Avaliar" ao lado do estagiário para o qual deseja registrar o feedback

E preencher o formulário com as informações necessárias

E clicar em "Cadastrar"

Então deve ser exibida uma mensagem de confirmação de que o feedback foi registrado com sucesso

E o feedback deve ser registrado no sistema, associado ao estagiário selecionado.

### Feedback Instrutor



Informação	Dado
ld Feedback	303
Nome	MARIAN STARK REILLY V
Trilha	QA
Email	mariaala@dbccompany.com.br
Cargo Responsável	Instrutor
Status	Finalizado
Responsável	Gabriel Kleiman
Tipo Feedback	Positivo
Nota	99
Descrição	sadasdasdass

•

### 12. Relatórios de Testes Automatizados

### TESTES DA API - FEATURE PROGRAMA

CENÁRIO: Falha ao editar programa erro ao enviar datas ou/e status inválidos

Dado que seja passado token válid

E que o contentType seja JSON

Quando definir valor das chaves "dataInicio" ou "dataFim" ou "status" como inválidos (fora do regex)

Então retorna status code 404

E retorna body JSON com chaves "timestamp", "status" e "errors"

Link:

CENÁRIO: Falha ao deletar programa com id inexistente

Dado que o endpoint é "/programa/delete/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 404

E mensagem de erro informando que o programa não foi encontrado.

CENÁRIO: Falha ao deletar programa com letras no id

Dado que o endpoint é "/programa/delete/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" com letras

Quando requisitar DELETE

Então a resposta deve conter status code 400

E mensagem de erro informando que o id deve ser numérico.

### CENÁRIO: Deletar programa com sucesso

Dado que o endpoint é "/programa/delete/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" existente

Quando requisitar DELETE

Então a resposta deve conter status code 200

E mensagem de sucesso informando que o programa foi deletado com sucesso.

#### CENÁRIO: Falha ao desativar programa com id inexistente

Dado que o endpoint é "/programa/deactivate/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 400 (ACEITÁVEL 404)

### CENÁRIO: Falha ao desativar programa com letras no id

Dado que o endpoint é "/programa/deactivate/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inválido com letras

Ouando requisitar DELETE

Então a resposta deve conter status code 404

### CENÁRIO: Desativar programa com sucesso

Dado que o endpoint é "/programa/deactivate/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" existente

Quando requisitar DELETE

Então a resposta deve conter status code 201

# CENÁRIO: Falha ao editar programa com id inexistente



Dado que o endpoint é "/programa/update/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" inexistente

E o body JSON contém novas informações válidas do programa

Quando requisitar PUT

Então a resposta deve conter status code 404

E mensagem de erro informando que o programa não foi encontrado.

(ACEITÁVEL MENSAGEM FORA DO PADRÃO)

Link:

### CENÁRIO: Falha ao editar programa com id vazio

Dado que o endpoint é "/programa/update/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" vazio

E o body JSON contém novas informações válidas do programa

Quando requisitar PUT

Então a resposta deve conter status code 400 (ACEITAVEL 404)

Link:

CENÁRIO: CT04: Editar programa com sucesso

Dado que o endpoint é "/programa/update/{idPrograma}"

E usuário está autenticado como administrador ou GP

E o path contém "idPrograma" existente

E o body JSON contém novas informações válidas do programa

Quando requisitar PUT

Então a resposta deve conter status code 200

E body em JSON com informações atualizadas do programa.

CENÁRIO: Falha ao criar um programa com datas abaixo da atual

Dado que o endpoint é "/programa/create"

E usuário está autenticado como administrador ou GP

E o body contém informações válidas do programa a ser criado, exceto pela data de início e data de fim abaixo da data atual

Quando requisitar POST

Então a resposta deve conter status code 400

E mensagem de erro informando que as datas de início e fim devem ser posteriores à data atual.

CENÁRIO: Falha ao criar um programa com nome nulo

Dado que o endpoint é "/programa/create"

E usuário está autenticado como administrador ou GP

E o body contém informações válidas do programa a ser criado, exceto pelo nome nulo

Quando requisitar POST

Então a resposta deve conter status code 400

E mensagem de erro informando que o nome é obrigatório.

CENÁRIO: Criar programa com sucesso

Dado que o endpoint é "/programa/create"

E usuário está autenticado como administrador ou GP

E o body contém informações válidas do programa a ser criado

Quando requisitar POST

Então a resposta deve conter status code 201

E body em JSON com informações do programa criado.

#### FEATURE TRILHA

CENÁRIO: Falha uscar todas trilhas

Dadoque eu tenho trilhas cadastradas no sistema

Quando nã informo o numero de paginas

E nem o tamanho

Então me retorna um erro 400

CENÁRIO: Listar todas as trilhas

Dado que eu tenho trilhas cadastradas no sistema

Quando informo o numero de paginas

E o tamanho

Então me retorna as trilhas cadastradas

CENÁRIO: Falha ao criar trilha com modulo Sem ID

Dadoque eu crio um programa

Ouando crio uma trilha

E não informo o id do modulo criado

Então recebo uma mensagem "idPrograma: must not be null"

CENÁRIO: Criar uma trilha com modulo com sucesso

Dadoque eu crio um programa

Quando crio uma trilha E crio junto um modulo Então a trilha é cadastrada com um módulo

CENÁRIO: Falha ao deletar trilha com id inexistente Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id da trilha inexistente Então não consigo deletar a trilha

CENÁRIO: Deletar trilha pelo id da trilha Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id da trilha Então a trilha é deletada com sucesso

CENÁRIO: Falha ao desativar trilha com id inexistente Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id da trilha inexistente Então não consigo deletar a trilha

CENÁRIO: Desativar trilha pelo id da trilha Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id da trilha Então a trilha é deletada com sucesso

CENÁRIO: Falha ao listar trilha sem o id Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id do programa com um valor vazio Então me retorna uma mensagem de erro de status 400

CENÁRIO: Listar trilha pelo id da trilha Dado que eu tenho uma trilha cadastradas no sistema Quando eu informo o id da trilha Então me retorna a trilha cadastrada

CENÁRIO: Falha ao listar todas trilhas com id negativo Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id do programa com um valor negativo Então me retorna uma mensagem de erro de status 400

CENÁRIO: Listar todas trilhas pelo id do programa Dado que eu tenho trilhas cadastradas no sistema Quando eu informo o id do programa relacionado a trilha Então me retorna a trilha cadastrada no programa

CENÁRIO: Falha ao atualizar uma trilha sem Body Dado que eu tenho uma trilha cadastrada Quando acesso o endpoint de atualizar trilha com id da trilha valido E não informo o corpo da requisição Então me retorna uma mensagem de erro de status 400

CENÁRIO: Falha ao atualizar uma trilha sem ID Dado que eu tenho uma trilha cadastrada Quando acesso o endpoint de atualizar trilha sem informar o id Então io sistema me retorna um erro E recebo a mensagem "idPrograma: must not be null"

CENÁRIO: Atualizar uma trilha com sucesso Dado que eu tenho uma trilha cadastrada Quando acesso o endpoint de atualizar trilha com id da trilha valida Então a trilha recebe novos dados

CENÁRIO: Tentar criar trilha com id inexistente Dado que eu tenho uma trilha com id invalido Quando eu tento adicionar uma trilha

Então devo receber uma mensagem escrito "status: campo invalido!"

CENÁRIO: Tentar adicionar trilha sem id de programa Dado que eu tenho uma trilha sem o campo de id do programa

Quando eu tentar adicionar a trilha

Então devo receber uma resposta de erro com o código de status 400

CENÁRIO: Criar trilia com sucesso Dado que eu tenha um programa criado Quando eu adiciono uma nova trilha Então a trilha deve ser cadastrada com sucesso

#### FEATURE ACOMPANHAMENTO

CENÁRIO: Falha em desativar um acompanhamento com id inexistente

Dado que eu acesso o campo de desativar acompanhamento

Quando informo o id do acompanhamento inexistente

Então ele retorna a mensagem "delete.id Acompanhamento: must be greater than or equal to 1" e o status code 400

CENÁRIO: Desativar um acompanhamento com sucesso Dado que eu acesso o campo de desativar acompanhamento

Quando informo o id do acompanhamento

Então ele deleta o acompanhamento com sucesso

CENÁRIO: Falha em deletar um acompanhamento com id inexistente

Dado que eu acesso o campo de delete

Quando informo o id do acompanhamento inexistente

Então ele retorna a mensagem "delete.idAcompanhamento: must be greater than or equal to 1" e o statuscode 400

CENÁRIO: Deletar um acompanhamento com sucesso

Dado que eu acesso o campo de delete Quando informo o id do acompanhamento

Então ele deleta o acompanhamento com sucesso

CENÁRIO: Falha ao buscar acompanhamento com id do acompanhamento inexistente

Dado que eu tenha um acomapnhamento criado

Quando informo o id do acompanhamento com uma numeração errada

Então ele me retorna a mensagem "getById.idAcompanhamento: must be greater than or equal to 1" e o statuscode 400

CENÁRIO: Buscar acompanhamento pelo id do acompanhamento

Dado que eu tenha um acomapnhamento criado

Quando informo o id do acompanhamento

Então ele me retorna o acompanhamento com sucesso

CENÁRIO: Falha ao listar acompanhamento com numero de paginas invalido

Dado que eu tenha um acomapnhamento criado

Quando não informo o numero de paginas

E nem o tamanho

Então me é mostrado uma mensagem status code 400

CENÁRIO: Buscar toddos os acompanhamentos

Dado que eu tenha um acomapnhamento criado

Quando informo o numero de paginas

E o tamanho

Então me é mostrado uma lista de acompanhamentos

CENÁRIO: Falha ao listar acompanhamento com numero de id invalido

Dado que eu tenha um acomapnhamento criado

Quando não informo o id do programa

Então me é mostrado uma mensagem de status code 400

CENÁRIO: Listar acompanhamento pelo id do programa

Dado que eu tenha um acomapnhamento criado

Quando informo o id do programa

Então ele me retorna o acompanhamento cadastrado no programa

CENÁRIO: Falha ao criar acompanhamento sem id

Dado que eu tenha um programa criado mas não informo o id

E tenho uma trilha criada

Quando tento adicionar um acompanhamento

Então me retorna uma mensagem de status code 400

CENÁRIO: Criar acompanhamento com sucesso

Dado que eu tenha um programa criado

E tenho uma trilha criada

Quando adiciono um acompanhamento

Então o acompanhamento e cadastrado com sucesso

#### FEATURE ESTAGIARIO

CENARIO: Atualizar estagiário com Nome inválido

Cenário: Atualizar estagiário com Nome inválido

Dado que o path do endpoint para atualização de estagiários está definido

E existe um estagiário cadastrado com ID válido

E o body da requisição contém um campo "nome" inválido

Quando enviar uma requisição PUT com os dados do estagiário atualizados

Então a resposta deve conter status code 400

E o body em JSON deve conter os atributos "timestamp", "status" e "erros" com as informações correspondentes ao erro de validação do campo "nome".

CENARIO: Desativar estagiário por ID inexistente

Dado que o endpoint é "/estagiario/deactivate/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 404

CENARIO: Desativar estagiário já desativado

Dado que o endpoint é "/estagiario/deactivate/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o estagiário já está desativado

Quando requisitar DELETE

Então a resposta deve conter status code 404

#### CENARIO: Desativar estagiário

Dado que o endpoint é "/estagiario/deactivate/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

Quando requisitar DELETE

Então a resposta deve conter status code 200

E body JSON com dados do usuário com atributo "ativo" igual a "false".

#### CENARIO: Deletar estagiário inexistente

Dado que o endpoint é "/estagiario/delete/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

Quando requisitar DELETE

Então a resposta deve conter status code 404.

#### CENARIO: Deletar estagiário com sucesso

Dado que o endpoint é "/estagiario/delete/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

Quando requisitar DELETE

Então a resposta deve conter status code 204.

### CENARIO: Atualizar estagiário com CPF inválido

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado, exceto pelo CPF inválido

Quando requisitar PUT

Então a resposta deve conter status code 400

E mensagem de erro informando que o CPF é inválido.

### CENARIO: Atualizar estagiário com Email Corporativo inválido

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado, exceto pelo email corporativo inválido

Quando requisitar PUT

Então a resposta deve conter status code 400

E mensagem de erro informando que o email corporativo é inválido.

### CENARIO: Atualizar estagiário com Email Pessoal inválido

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado, exceto pelo email pessoal inválido

Quando requisitar PUT

Então a resposta deve conter status code 400

E mensagem de erro informando que o email pessoal é inválido.

#### CENARIO: Atualizar estagiário inexistente

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" inexistente

E o body contém informações válidas do estagiário a ser atualizado

Quando definir body JSON com chaves válidas do estagiário

E requisitar PUT

Então a resposta deve conter status code 404

E mensagem de erro informando que o estagiário não foi encontrado.

#### CENARIO: Atualizar estagiário com sucesso

Dado que o endpoint é "/estagiario/update/{idEstagiario}"

E usuário está autenticado

E o path contém "idEstagiario" existente

E o body contém informações válidas do estagiário a ser atualizado

Quando definir body JSON com chaves válidas do estagiário

E requisitar PUT

Então a resposta deve conter status code 200

E body em JSON com informações atualizadas do estagiário.

### CENARIO: Buscar por lista de todos estagiários com página e tamanho inválidos

Dado que o endpoint é "/estagiario/list-all"

E usuário esta autenticado

E o query contem "pagina" com valor inválidos

E o query contem "tamanho" com valor inválidos

Quando requisitar GET

Então a resposta deve conter status code 400.

#### CENARIO: Buscar por lista de todos estagiários

Dado que o endpoint é "/estagiario/list-all"

E usuário esta autenticado

E o query contem "pagina" com número maior ou igual a zero

E o query contem "tamanho" com número maior ou igual a um

Quando requisitar GET

Então a resposta deve conter status code 200

E body em JSON com número de páginas informado

E tamanho informado

E estagiários nesta página e tamanho.

#### CENARIO: Buscar estagiário por programa com página e tamanho inválidos

**Dado** que o endpoint é "/estagiario/list-by-programa/{idPrograma}"

E usuário esta autenticado

E o path contem "idPrograma" existente

E o query contem "pagina" com valor inválido

E o query contem "tamanho" com valor inválido

Quando requisitar GET

Então a resposta deve conter status code 400.

#### CENARIO: Buscar estagiário por programa

Dado que o endpoint é "/estagiario/list-by-programa/{idPrograma}"

E usuário esta autenticado

E o path contem "idPrograma" existente

E o query contem "pagina" com número maior ou igual a zero

E o query contem "tamanho" com número maior ou igual a um

Quando requisitar GET

Então a resposta deve conter status code 200

E body em JSON com número de páginas informado

E tamanho informado

E estagiários pertencentes ao programa nesta página e tamanho.

#### CENARIO: Buscar estagiário por ID inválido

**Dado** que o endpoint é "/estagiario/get-by-id/{idEstagiario}"

E usuário esta autenticado

E o path contem "idEstagiario" inválido (diferente de número maior que 0)

Quando requisitar GET

Então a resposta deve conter status code 400.

#### CENARIO: Buscar estagiário por ID inexistente Dado que o endpoint é "/estagiario/get-by-

id/{idEstagiario}"

E usuário esta autenticado

E o path contem "idEstagiario" inexistente

Quando requisitar GET

Então a resposta deve conter status code 404.

### CENARIO: Buscar estagiário por ID com sucesso

Dado que o endpoint é "/estagiario/get-by-id/{idEstagiario}"

E usuário esta autenticado

E o path contem "idEstagiario" existente

Quando requisitar GET

Então a resposta deve conter status code 200

E body em JSON com atributos do estagiário.

CENARIO: Criar Estagiário com Email Corporativo inválido

Dado que o path do endpoint está correto

E usuário esta autenticado

Quando definir body JSON com chave "emailCorporativo" inválido

E demais chaves válidas do estagiário

E requisitar POST

Então a resposta deve conter status code 400

E body em JSON com atributos com timestamp, status e erros respectivos.

CENARIO: Criar Estagiário com Email Pessoal inválido

Dado que o path do endpoint está correto

E usuário esta autenticado

Quando definir body JSON com chave "emailPessoal" inválido

E demais chaves válidas do estagiário

E requisitar POST

Então a resposta deve conter status code 400

E body em JSON com atributos com timestamp, status e erros respectivos.

CENARIO: Criar estagiário com CPF com ponto e hífen

Dado que o path do endpoint está correto

E usuário esta autenticado

Quando definir body JSON com chave "cpf" inválido ("071.545.751-90")

E demais chaves válidas do estagiário

E requisitar POST

Então a resposta deve conter status code 400

E body em JSON com atributos com timestamp, status e erros respectivos.

CENARIO: Criar estagiário com sucesso

Dado que o path do endpoint está correto

E usuário esta autenticado

Quando definir body JSON com chaves válidas do estagiário

Então a resposta deve conter status code 201

E body em JSON com atributos do estagiário.

# FEATURE AVALIAÇÃO

CENARIO: Erro ao desativar avaliação pelo id da avaliação

Dado que eu tenha uma avaliação criada

Quando eu adiciono o id avaliação invalido no campo de delete

Então é mostrada uum erro de status code 400

CENARIO: Desativar avaliação pelo id da avaliação

Dado que eu tenha uma avaliação criada

Quando eu adiciono o id da avaliação no campo de delete

Então a avaliação é excluida corretamente

CENARIO: Erro ao deletar avaliação pelo id da avaliação

Dado que eu tenha uma avaliação criada

Quando eu adiciono o id avaliação invalido no campo de delete

Então é mostrada uum erro de status code 400

CENARIO: Deletar avaliação pelo id da avaliação

Dado que eu tenha uma avaliação criada

Quando eu adiciono o id da avaliação no campo de delete

Então a avaliação é excluida corretamente

CENARIO: Erro ao todos os acompanhamentos

Dado que eu tenha uma mais de uma avaliação criada

Quando eu não adiciono o numero de paginas e o tamanho

Então é mostrada uum erro de status code 400

CENARIO: Buscar todos os acompanhamentos

Dado que eu tenha uma mais de uma avaliação criada

Quando eu adiciono o numero de paginas e o tamanho

Então é mostrada uma lista de avaliações

CENARIO: Buscar avaliação pelo id da avaliação

Dado que eu tenha uma avaliação criada

Quando eu adiciono o id da avaliação no campo de busca

Então a avaliação é xibida corretamente

CENARIO: Buscar avaliação pelo id do acompanhamento invalido

Dado que eu tenha um programa criado

Quando eu adiciono o id acompanhamento com um valor invalido no campo de busca

Então é me retorna o statuscode 400

CENARIO: Buscar avaliação pelo id do acompanhamento

Dado que eu tenha um programa criado

Quando eu adiciono o id acompanhamento no campo de busca

Então o acmopanhamento é xibido corretamente

CENARIO: Atualizar avaliação com sucesso

Dado que eu tenha uma avaliação criada

Quando eu adiciono novas informações para a avaliação

Então a avaliação deve ser atualizada com sucesso

CENARIO: Criar avaliação com sucesso

Dado que eu tenha um acompanhamento criado

E um estágiario cadastrado

Quando eu adiciono uma avaliação sem id do estagiario

Então o acmopanhamento deve ser cadastrado com sucesso

CENARIO: Criar avaliação com sucesso

Dado que eu tenha um acompanhamento criado

E um estágiario cadastrado

Quando eu adiciono uma nova avaliação

Então a avaliação deve ser cadastrado com sucesso

### FEATURE AGENDAMENTO

Cenário: DELETAR AGENDAMENTO PELO ID INEXISTENTE

DADO que eu informe o id de um agendamento inexistente

QUANDO colocoo o id para deletar o agendamento

ENTÃO me mostra a mensagem de statuscode 400, mostrando que o id não existe

Cenário: DELETAR AGENDAMENTO PELO ID

DADO que eu tenha um acompanhamento cadastrado

QUANDO colocoo id para deletarr um agendamento

ENTÃO me mostra o agendamento foi deletado do sistema

Cenário: BUSCAR TODOS OS AGENDAMENTOS SEM INFORMAI PAGINAS

DADO que eu tenha um ou mais acompanhamento cadastrado

QUANDO colocooo numero de paginas e tamanho ENTÃO me mostra a mensagem de statuscode 400

Cenário: BUSCAR TODOS OS AGENDAMENTOS

DADO que eu tenha um ou mais acompanhamento cadastrado

QUANDO colocooo numero de paginas e tamanho ENTÃO me mostra a lista de agendamentos cadastrados

Cenário: BUSCAR AGENDAMENTO PELO ID INEXISTENTE

DADO que eu tenha um acompanhamento cadastrado

QUANDO colocoo id inexistente para buscar um agendamento

ENTÃO me mostra a mensagem de statuscode 400, mostrando que o id não existe

Cenário: BUSCAR AGENDAMENTO PELO ID

DADO que eu tenha um acompanhamento cadastrado

QUANDO colocoo id para buscar um agendamento

ENTÃO me mostra o agendamento cadastrado no sistema

Cenário: ATUALIZAR AGENDAMENTO COM SUCESSO

DADO que eu tenha um agendamento cadastrado

QUANDO coloco as requisições no body para atualizar o agendamento

E informo o id do agendamento

ENTÃO atualizo um agendamento com sucesso

Cenário: GERAR AGENDAMENTO COM id errado

DADO que eu tenha um acompanhamento cadastrado

QUANDO coloco as requisições para criar um agendamento

E informo o id do acompanhamento inexistente

ENTÃO aparece uma mensagem de erro com status code 400

Cenário: GERAR AGENDAMENTO COM SUCESSO

DADO que eu tenha um acompanhamento cadastrado

QUANDO coloco as requisições para criar um agendamento

E informo o id do acompanhamento

ENTÃO cadastro um agendamento com sucesso

### FEATURE FEEDBACK

CENÁRIO: DELETAR FEEDBACK PELO ID DO FEEDBACK INVALIDO

DADO que eu tenha um agendamento um cadastrado

QUANDO informo um id de feedback invalido

ENTÃO me é informado que o feedback não existe

CENÁRIO: DELETAR FEEDBACK PELO ID DO FEEDBACK

DADO que eu tenha um agendamento um cadastrado

QUANDO informo o id do feedback

ENTÃO é deletado o feedback

CENÁRIO: BUSCAR FEEDBACK COM O ID DO AGENDAMENTO INVALIDA

DADO que eu tenha um agendamento um cadastrado

QUANDO informo o id de uma agendamento inexistente

ENTÃO me retorna o statuscode 400 informando que a avaliação é inválida

### CENÁRIO: BUSCAR FEEDBACK PELO ID DO AGENDAMENTO

DADO que eu tenha um agendamento um cadastrado

QUANDO informo o id do agendamento

ENTÃO me retorna o feedback cadastrado no agendamento

### CENÁRIO: BUSCAR FEEDBACK COM O ID DO FEEDBACK INVALIDO

DADO que eu tenha um agendamento um cadastrado

QUANDO informo o id do feedback inexistente

ENTÃO me retorna o statuscode 400 informando que o feedback é inválido

#### CENÁRIO: BUSCAR FEEDBACK PELO ID DO FEEDBACK

DADO que eu tenha um agendamento um cadastrado

QUANDO informo o id do feedback

ENTÃO me retorna o feedback cadastrado

### CENÁRIO: ATUALIZAR FEEDBACK COM ID DO AGENDAMENTO INVALIDO

DADO que eu tenha um agendamento um cadastrado

QUANDO adiciono os requisitos de um feedback

E informo o id do agendamento inexistente

ENTÃO me retorna o statuscode 400 informando que o id é invalido

### CENÁRIO: ATUALIZAR FEEDBACK

DADO que eu tenha um agendamento um cadastrado

QUANDO adiciono os requisitos para a atualisação de um feedback

ENTÃO me uma mensagem de erro statuscode 200 mostrando que o feedback foi atualizado

#### CENÁRIO: CRIAR FEEDBACK COM ID DO AGENDAMENTO INVALIDO

DADO que eu tenha um agendamento um cadastrado

QUANDO adiciono os requisitos de um feedback

E informo o id do agendamento inexistente

ENTÃO me retorna o statuscode 400 informando que o id é invalido

#### -CENÁRIO: CRIAR FEEDBACK

DADO que eu tenha um agendamento um cadastrado

QUANDO adiciono os requisitos de um feedback

E informo o id do agendamento e

ENTÃOme retorna o feedback