

Documento de Testes

API Jornada Milhas

Sumário

- 1) Suíte de Testes Destinos: 3 – 23
- 2) Suíte de Testes Depoimentos: 24 - 41

Caso de teste

Id: 1

Suíte: Destinos

Método: Post

Resumo: Ao adicionar um destino com sucesso a api deve retornar código 201.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente.

Resultados esperados:

- 1) Receber código 201 como resposta;
- 2) Receber um JSON com os dados do objeto criado.

Caso de teste

Id: 2

Suíte: Destinos

Método: Post

Resumo: Ao adicionar um destino sem o campo "nome" a api deve retornar código 400 com a seguinte mensagem "O nome do destino é obrigatório".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'nome' que deve estar vazio.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a mensagem de erro: "O nome do destino é obrigatório".

Caso de teste

Id: 3

Suíte: Destinos

Método: Post

Resumo: Ao Adicionar um destino com o campo "foto_1" vazio a api deve retornar código 201 e o objeto criado deve receber os dados gerados pelo método GerarImagem” da integração com ChatGPT.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint ‘/Destinos’;
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo ‘foto_1’ que deve estar vazio.

Resultados esperados:

- 1) Receber código 201 como resposta;
- 2) Receber um JSON com os dados do objeto criado;
- 3) O campo ‘foto_1’ deve possuir os dados gerados pelo método ‘GerarImagem’.

Caso de teste

Id: 4

Suíte: Destinos

Método: Post

Resumo: Ao Adicionar um destino com o campo "foto_2" vazio a api deve retornar código 201 e o objeto criado deve receber os dados gerados pelo método GerarImagem” da integração com ChatGPT.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint ‘/Destinos’;
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo ‘foto_2’ que deve estar vazio.

Resultados esperados:

- 1) Receber código 201 como resposta;
- 2) Receber um JSON com os dados do objeto criado;
- 3) O campo ‘foto_2’ deve possuir os dados gerados pelo método ‘GerarImagem’.

Caso de teste

Id: 5

Suíte: Destinos

Método: Post

Resumo: Ao Adicionar um destino com o campo "meta" vazio a api deve retornar código 400 com a seguinte mensagem "O destino deve possuir uma meta".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'meta' que deve estar vazio.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a mensagem de erro: "O destino deve possuir uma meta".

Caso de teste

Id: 6

Suíte: Destinos

Método: Post

Resumo: Ao Adicionar um destino com o campo "texto_descritivo" vazio a api deve retornar código 201 e o objeto criado deve receber os dados gerados pelo método GerarImagem da integração com ChatGPT.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'texto_descritivo' que deve estar vazio.

Resultados esperados:

- 1) Receber código 201 como resposta;
- 2) Receber um JSON com os dados do objeto criado;
- 3) O campo 'texto_descritivo' deve possuir os dados gerados pelo método 'GerarImagem'.

Caso de teste

Id: 7

Suíte: Destinos

Método: Post

Resumo: Ao adicionar um destino com o campo "nome" com menos de 5 caracteres a api deve retornar código 400 com a seguinte mensagem "O nome do destino deve possuir no mínimo 5 caracteres".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'nome' que deve possuir menos de 5 caracteres.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a mensagem de erro: "O nome do destino deve possuir no mínimo 5 caracteres".

Caso de teste

Id: 8

Suíte: Destinos

Método: Post

Resumo: Ao Adicionar um destino com o campo "nome" com mais de 255 caracteres a api deve retornar código 400 com a seguinte mensagem "O nome do destino deve possuir no máximo 255 caracteres".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'nome' que deve possuir mais de 255 caracteres.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a mensagem de erro: "O nome do destino deve possuir no máximo 255 caracteres".

Caso de teste

Id: 9

Suíte: Destinos

Método: Post

Resumo: Ao Adicionar um destino com o campo "meta" com mais de 160 caracteres a api deve retornar código 400 com a seguinte mensagem "A meta deve possuir no máximo 160 caracteres".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Destinos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'meta' que deve possuir mais de 160 caracteres.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a mensagem de erro: "A meta deve possuir no máximo 160 caracteres".

Caso de teste

Id: 10

Suíte: Destinos

Método: Get

Resumo: Ao buscar todos os destinos sem uso de filtros a api deve retornar código 200 e uma lista de destinos

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Destinos'.

Resultados esperados:

- 1) Receber código 200 como resposta;
- 2) Receber um JSON com uma lista de objetos do tipo 'Destino'.

Caso de teste

Id: 11

Suíte: Destinos

Método: Get

Resumo: Ao buscar destinos com um nome solicitado a api deve retornar código 200 e uma lista de destinos que contenham esse nome.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o nome a ser pesquisado.

Resultados esperados:

- 1) Receber código 200 como resposta;
- 2) Receber um JSON com uma lista de objetos do tipo 'Destino';
- 3) Todos os objetos deste JSON devem possuir no parâmetro 'nome' o nome que foi utilizado na url de pesquisa.

Caso de teste

Id: 12

Suíte: Destinos

Método: Get

Resumo: Ao buscar destinos com um nome específico a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado" caso nenhum destino seja encontrado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição um nome que não exista no banco de dados para ser pesquisado.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 13

Suíte: Destinos

Método: Get

Resumo: Ao buscar um destino por id a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado" caso nenhum destino seja encontrado com o id informado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição um id que não exista no banco de dados para ser pesquisado.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 14

Suíte: Destinos

Método: Get

Resumo: Ao buscar um destino por id a api deve retornar código 200 e um JSON com as informações daquele destino caso um destino seja encontrado com o id informado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto a ser pesquisado.

Resultados esperados:

- 1) Receber código 200 como resposta;
- 2) Receber um JSON com os dados do objeto pesquisado;

Caso de teste

Id: 15

Suíte: Destinos

Método: Put

Resumo: Ao tentar atualizar um destino informando um id que não possui cadastro a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PUT para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto ser atualizado;
- 3) Enviar o payload com os dados do destino preenchidos corretamente.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 16

Suíte: Destinos

Método: Put

Resumo: Ao tentar atualizar um destino informando um id que possui cadastro a api deve retornar código 204.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PUT para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto ser atualizado;
- 3) Enviar o payload com os dados do destino preenchidos corretamente.

Resultados esperados:

- 1) Receber código 204 como resposta;
- 2) Receber um JSON com os dados do objeto atualizado.

Caso de teste

Id: 17

Suíte: Destinos

Método: Patch

Resumo: Ao tentar atualizar parcialmente um destino informando um id que possui cadastro o sistema deve retornar código 204.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PATCH para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto ser atualizado;
- 3) Enviar o payload com os dados do destino preenchidos corretamente.

Resultados esperados:

- 1) Receber código 204 como resposta;
- 2) Receber um JSON com os dados do objeto atualizado.

Caso de teste

Id: 18

Suíte: Destinos

Método: Patch

Resumo: Ao tentar atualizar parcialmente um destino informando um id que não possui cadastro a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PATCH para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição um id que não exista no banco de dados;
- 3) Enviar o payload com os dados do destino preenchidos corretamente.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 19

Suíte: Destinos

Método: Patch

Resumo: Ao tentar atualizar parcialmente um destino informando um campo que não existe a api deve retornar código 400.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PATCH para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto ser atualizado.
- 3) Enviar o payload com um campo que não exista para ser atualizado.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 20

Suíte: Destinos

Método: Delete

Resumo: Ao tentar deletar um destino informando um id que não possui cadastro a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método DELETE para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto ser deletado.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 21

Suíte: Destinos

Método: Delete

Resumo: Ao tentar deletar um destino informando um id que possui cadastro a api deve retornar código 204.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método DELETE para o endpoint '/Destinos';
- 2) Enviar como parâmetro na url de requisição o id do objeto ser deletado.

Resultados esperados:

- 1) Receber código 204 como resposta;

Caso de teste

Id: 22

Suíte: Depoimentos

Método: Post

Resumo: Ao adicionar um depoimento com sucesso a api deve retornar código 201.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Depoimentos';
- 2) Enviar o payload com todos os dados preenchidos corretamente.

Resultados esperados:

- 1) Receber código 201 como resposta;

Caso de teste

Id: 23

Suíte: Depoimentos

Método: Post

Resumo: Ao adicionar um depoimento sem o campo "nome" a api deve retornar código 400 com a seguinte mensagem " O nome do criador do depoimento é obrigatório".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Depoimentos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'nome' que deve estar vazio.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a seguinte mensagem de erro: "O nome do criador do depoimento é obrigatório".

Caso de teste

Id: 24

Suíte: Depoimentos

Método: Post

Resumo: Ao adicionar um depoimento com o campo "nome" com menos de 10 caracteres a api deve retornar código 400 com a seguinte mensagem "O nome do criador deve possuir no mínimo 10 caracteres".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Depoimentos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'nome' que deve possuir menos de 10 caracteres.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a seguinte mensagem de erro: "O nome do criador deve possuir no mínimo 10 caracteres".

Caso de teste

Id: 25

Suíte: Depoimentos

Método: Post

Resumo: Ao Adicionar um depoimento com o campo "nome" com mais de 255 caracteres a api deve retornar código 400 com a seguinte mensagem "O nome do criador deve possuir no máximo 255 caracteres".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Depoimentos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'nome' que deve possuir mais de 255 caracteres.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a seguinte mensagem de erro: "O nome do criador deve possuir no máximo 255 caracteres".

Caso de teste

Id: 26

Suíte: Depoimentos

Método: Post

Resumo: Ao Adicionar um depoimento com o campo "depoimento" vazio a api deve retornar código 400 com a seguinte mensagem "O depoimento não pode estar em branco".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Depoimentos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'depoimento' que deve estar vazio.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a seguinte mensagem de erro: "O depoimento não pode estar em branco".

Caso de teste

Id: 27

Suíte: Depoimentos

Método: Post

Resumo: Ao adicionar um depoimento com o campo "depoimento" com menos de 10 caracteres a api deve retornar código 400 com a seguinte mensagem "O depoimento deve possuir no mínimo 10 caracteres".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método POST para o endpoint '/Depoimentos';
- 2) Enviar o payload com todos os dados preenchidos corretamente exceto o campo 'depoimento' que deve possuir menos de 10 caracteres.

Resultados esperados:

- 1) Receber código 400 como resposta;
- 2) Receber um JSON com a seguinte mensagem de erro: "O depoimento deve possuir no mínimo 10 caracteres".

Caso de teste

Id: 28

Suíte: Depoimentos

Método: Get

Resumo: Ao buscar todos os depoimentos sem uso de filtros a api deve retornar código 200 e uma lista de destinos.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Depoimentos'.

Resultados esperados:

- 1) Receber código 200 como resposta;
- 2) Receber um JSON com uma lista de objetos do tipo 'Depoimento'.

Caso de teste

Id: 29

Suíte: Depoimentos

Método: Get

Resumo: Ao buscar depoimentos com um nome solicitado a api deve retornar código 200 e uma lista de depoimentos que contenham esse nome.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição o nome a ser pesquisado.

Resultados esperados:

- 1) Receber código 200 como resposta;
- 2) Receber um JSON com uma lista de objetos do tipo 'Depoimento';
- 3) Todos os objetos deste JSON devem possuir no parâmetro 'nome' o nome que foi utilizado na url de pesquisa.

Caso de teste

Id: 30

Suíte: Depoimentos

Método: Get

Resumo: Ao buscar depoimentos com um nome específico a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado" caso nenhum destino seja encontrado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição um nome que não exista no banco de dados para ser pesquisado.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 31

Suíte: Depoimentos

Método: Get

Resumo: Ao buscar um depoimento por id a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado" caso nenhum destino seja encontrado com o id informado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição um id que não exista no banco de dados para ser pesquisado.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 32

Suíte: Depoimentos

Método: Get

Resumo: Ao buscar um depoimento por id a api deve retornar código 200 e um JSON com as informações daquele destino caso um destino seja encontrado com o id informado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método GET para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição um id que exista no banco de dados para ser pesquisado.

Resultados esperados:

- 1) Receber código 200 como resposta;
- 2) Receber um JSON com os dados do objeto que foi pesquisado.

Caso de teste

Id: 33

Suíte: Depoimentos

Método: Put

Resumo: Ao tentar atualizar um depoimento informando um id que não possui cadastro a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PUT para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição um id que não exista no banco de dados para ser pesquisado;
- 3) Enviar um JSON com os dados a serem atualizados do objeto no corpo da requisição.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 34

Suíte: Depoimentos

Método: Put

Resumo: Ao tentar atualizar um depoimento informando um id que possui cadastro o sistema deve retornar código 204 e um JSON validando que o Depoimento foi alterado.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PUT para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição um id que exista no banco de dados para ser pesquisado;
- 3) Enviar um JSON com os dados a serem atualizados do objeto no corpo da requisição.

Resultados esperados:

- 1) Receber código 204 como resposta;
- 2) Receber um JSON com os dados do objeto que foi atualizado.

Caso de teste

Id: 35

Suíte: Depoimentos

Método: Patch

Resumo: Ao tentar atualizar parcialmente um depoimento informando um id que possui cadastro o sistema deve retornar código 204.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PATCH para o endpoint `/Depoimentos`;
- 2) Enviar como parâmetro na url de requisição um id que exista no banco de dados para ser pesquisado;
- 3) Enviar um JSON com o campo e os dados a serem atualizados do objeto no corpo da requisição.

Resultados esperados:

- 1) Receber código 204 como resposta;
- 2) Receber um JSON com os dados do objeto que foi atualizado.

Caso de teste

Id: 36

Suíte: Depoimentos

Método: Patch

Resumo: Ao tentar atualizar parcialmente um depoimento informando um id que não possui cadastro a api deve retornar código 404 e a mensagem "Nenhum resultado foi encontrado".

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PATCH para o endpoint `/Depoimentos`;
- 2) Enviar como parâmetro na url de requisição um id que não exista no banco de dados para ser pesquisado;
- 3) Enviar um JSON com o campo e os dados a serem atualizados do objeto no corpo da requisição.

Resultados esperados:

- 1) Receber código 404 como resposta;
- 2) Receber um JSON com a seguinte mensagem de erro: "Nenhum resultado foi encontrado".

Caso de teste

Id: 37

Suíte: Depoimentos

Método: Patch

Resumo: Ao tentar atualizar parcialmente um destino informando um campo que não existe a api deve retornar código 400.

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método PATCH para o endpoint `/Depoimentos`;
- 2) Enviar como parâmetro na url de requisição um id que exista no banco de dados para ser pesquisado;
- 3) Enviar um JSON com um campo inválido e os dados a serem atualizados do objeto no corpo da requisição.

Resultados esperados:

- 1) Receber código 400 como resposta;

Caso de teste

Id: 39

Suíte: Depoimentos

Método: Delete

Resumo: Ao tentar deletar um depoimento informando um id que possui cadastro a api deve retornar código 204

Status: Aprovado

Pré-condição: Ter uma versão de testes da API Jornada Milhas em execução.

Passos:

- 1) Criar uma requisição usando método DELETE para o endpoint '/Depoimentos;
- 2) Enviar como parâmetro na url de requisição um id que exista no banco de dados para ser pesquisado;

Resultados esperados:

- 1) Receber código 204 como resposta;