

RT - Tulio

1. Cenários de Testes

Cenário 1 - Pessoas - Cadastrar/Editar/Listar		
Objetivo do Teste	<ol style="list-style-type: none">1. Cadastrar uma pessoa2. Listar todas as pessoas3. Listar uma pessoa específica4. Editar pessoa cadastrada5. Validar mensagens de alerta	
Critério de Aceite	<ol style="list-style-type: none">1. Cadastro realizado corretamente2. Pessoas listadas corretamente3. Edição realizada4. Mensagens apresentadas corretamente	
Pré-condições	<ol style="list-style-type: none">1. Postman configurado2. API configurada	
Passo	Ação do Passo	Resultados Esperados
1	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma pessoa:</p> <ul style="list-style-type: none">• Sem informar dados no campo [Nome] <p>Método: Post</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body:</p> <pre>{ "nome": EM BRANCO "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": true, "meta": 1000.00}</pre>	<p>sistema deverá validar o JSON e retornar um response informando que o campo NOME é obrigatório</p>

2	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma pessoa:</p> <ul style="list-style-type: none"> Sem informar dados no campo [dataNascimento] <p>Método: Post</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body:</p> <pre>{ "nome": "João da Silva", "dataNascimento": "EM BRANCO", "cpf": "464.589.520-66", "ativo": true, "meta": 1000.00 }</pre>	<p>sistema deverá validar o JSON e retornar um response informando que o campo data Nascimento é obrigatório</p>
3	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma pessoa:</p> <ul style="list-style-type: none"> Sem informar dados no campo [ativo] <p>Método: Post</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body:</p> <pre>{ "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": EM BRANCO, "meta": 1000.00 }</pre>	<p>sistema deverá validar o JSON e retornar um response informando que o campo ativo é obrigatório</p>

4	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma pessoa:</p> <ul style="list-style-type: none"> Sem informar dados no campo [meta] <p>Método: Post</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body:</p> <pre>{ "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": true, "meta": EM BRANCO }</pre>	<p>sistema deverá validar o JSON e retornar um response informando que o campo meta é obrigatório</p>
5	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma pessoa:</p> <ul style="list-style-type: none"> Informar dados inválidos no campo [CPF] <p>Método: Post</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body:</p> <pre>{ "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": true, "meta": EM BRANCO }</pre>	<p>sistema deverá validar o JSON e retorna o response:</p> <pre>{ "codigoErro": 2, "mensagemErro": "O cpf informado não é valido" }</pre>

<p>6</p>	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma pessoa:</p> <ul style="list-style-type: none"> • Informar todos os campos corretamente <p>Método: Post</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body:</p> <pre>{ "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": true, "meta": 1000.00 }</pre>	<p>sistema deverá validar o JSON e retorna o response:</p> <pre>{ "id": 1, "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": true, "meta": 1000 }</pre>
<p>7</p>	<p>Enviar na Ferramenta Postman o JSON com o Método PUT para editar o registro de uma pessoa:</p> <ul style="list-style-type: none"> • Modificar algum dado do body encaminhado <p>Método: PUT</p> <p>API: localhost:8080/api/pessoas</p> <p>Header: Content-Type: application/json</p> <p>Body (cadastrado)</p> <pre>{ "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "464.589.520-66", "ativo": true, "meta": 1000.00 }</pre> <p>Body: (Editado)</p> <pre>{ "id": 1, "nome": "João da Silva dos Santos", "dataNascimento": "01/01/2000", "cpf": "789.123.999/00", "ativo": false, "meta": 0 }</pre>	<p>sistema deverá validar o JSON e retorna o response:</p> <pre>{ "id": 1, "nome": "João da Silva dos Santos", "dataNascimento": "01/01/2000", "cpf": "789.123.999/00", "ativo": false, "meta": 0 }</pre>

8	<p>Enviar na Ferramenta Postman o Método GET para listar todas as pessoas</p> <p>Método: GET</p> <p>API: localhost:8080/api/pessoas?pagina=0&tamanho=10</p> <p>Header: Content-Type: application/json</p> <p>Os Parâmetros Página e Tamanho são utilizados para controlar os registros de retorno</p>	<p>sistema deverá validar o JSON e retorna o response com todas as pessoas cadastradas no sistema</p> <pre>{ "pagina": 0, "tamanho": 10, "numeroRegistros": 3, "registros": [{ "id": 1, "nome": "João da Silva", "dataNascimento": "2000-01-01", "cpf": "123.456.789/00", "ativo": true, "meta": 1000 }, { "id": 2, "nome": "Maria da Silva", "dataNascimento": "1998-01-01", "cpf": "987.654.321/00", "ativo": true, "meta": 1000 }, { "id": 3, "nome": "José da Silva", "dataNascimento": "2010-09-27", "cpf": null, "ativo": false, "meta": 0.5 }] }</pre>
8	<p>Enviar na Ferramenta Postman o Método GET para listar uma pessoa específica</p> <p>Método: GET</p> <p>API: localhost:8080/api/pessoas/1</p> <p>Header: Content-Type: application/json</p> <p>Obs: Neste caso estamos listando a pessoa com o id=1</p>	<p>sistema deverá validar o JSON e Retorna apenas as informações de uma pessoa, conforme solicitado</p> <pre>{ "id": 1, "nome": "João da Silva", "dataNascimento": "01/01/2000", "cpf": "464.589.520-66", "ativo": true, "meta": 1000 }</pre>
9	<p>Enviar na Ferramenta Postman o Método GET para listar uma pessoa específica inexistente</p> <p>Método: GET</p> <p>API: localhost:8080/api/pessoas/80</p> <p>Header: Content-Type: application/json</p> <p>Obs: Neste caso estamos listando a pessoa com o id=80</p>	<p>sistema deverá validar o JSON e Retorna reponse:</p> <pre>{ "codigoErro": 1, "mensagemErro": "O registro 80 não foi encontrado" }</pre>

Cenário 2 - Receitas - Cadastrar/Editar/Deletar/Listar

Objetivo do Teste	<ol style="list-style-type: none">1. Cadastrar uma receita2. Listar todas as receitas3. Listar uma receita específica4. Editar receita cadastrada5. Validar mensagens de alerta6. Deletar receita	
Critério de Aceite	<ol style="list-style-type: none">1. Cadastro realizado corretamente2. receita listada corretamente3. Edição realizada4. Mensagens apresentadas corretamente5. Receita deletada	
Pré-condições	<ol style="list-style-type: none">1. Postman configurado2. API configurada	
Passo	Ação do Passo	Resultados Esperados
1	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma receita:</p> <ul style="list-style-type: none">• Sem informar dados no campo [pessoald] <p>Método: Post</p> <p>API: localhost:8080/api/receitas</p> <p>Body:</p> <pre>{ "pessoald": EM BRANCO "data": "2021-02-15", "valor": 2500.00 }</pre>	sistema deverá validar o JSON e retornar um response informando que o campo pessoald é obrigatório
2	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma receita:</p> <ul style="list-style-type: none">• Sem informar dados no campo [data] <p>Método: Post</p> <p>API: localhost:8080/api/receitas</p> <p>Body:</p> <pre>{ "pessoald": 1, "data": EM BRANCO "valor": 2500.00 }</pre>	sistema deverá validar o JSON e retornar um response informando que o campo data é obrigatório

3	<p>Enviar na Ferramenta Postman o JSON com o Método POST para o registro de uma receita:</p> <ul style="list-style-type: none"> informar um valor negativo no campo [valor] <p>Método: Post</p> <p>API: localhost:8080/api/receitas</p> <p>Body:</p> <pre>{ "pessoaId": 1, "data": "2021-02-15", "valor": -10 }</pre>	<p>sistema deverá validar o JSON e retorna um response</p> <pre>{ "codigoErro": 3, "mensagemErro": "O campo valor não pode ser negativo" }</pre>
4	<p>Enviar na Ferramenta Postman o JSON com o Método PUT para editar o registro de uma receita:</p> <ul style="list-style-type: none"> Modificar algum dado do body encaminhado <p>Método: PUT</p> <p>API: localhost:8080/api/receitas/1</p> <p>Body Cadastrado:</p> <pre>{ "pessoaId": 1, "data": "2021-02-15", "valor": 2500.00 }</pre> <p>Body Editado</p> <pre>{ "pessoaId": 1, "data": "2021-02-15", "valor": 2537.5 }</pre>	<p>sistema deverá validar o JSON e retorna o response:</p> <pre>{ "id": 1, "pessoaId": 1, "data": "2021-02-15", "valor": 2537.5 }</pre>

5	<p>Enviar na Ferramenta Postman o JSON com o Método PUT para editar o registro de uma receita inexistente:</p> <ul style="list-style-type: none"> • Modificar algum dado do body • Incluir ID inexistente <p>Método: PUT</p> <p>API: localhost:8080/api/receitas/15</p> <p>Body:</p> <pre>{ "pessoald": 15, "data": "2021-02-15", "valor": 2500.00 }</pre>	<p>sistema deverá validar o JSON e retorna o response:</p> <pre>{ "codigoErro": 1, "mensagemErro": "O registro 15 não foi encontrado" }</pre>
6	<p>Enviar na Ferramenta Postman o Método GET para retornar uma lista de receitas</p> <p>Método: GET</p> <p>API: localhost:8080/api/receitas?pagina=0&tamanho=10</p> <p>Os Parâmetros Página e Tamanho são utilizados para controlar os registros de retorno</p>	<p>sistema deverá validar o JSON e retorna o response com lista de receitas cadastradas no sistema</p> <pre>{ "pagina": 0, "tamanho": 10, "numeroRegistros": 2, "registros": [{ "id": 1, "pessoald": 1, "data": "2021-01-12", "valor": 1100.99 }, { "id": 2, "pessoald": 1, "data": "2021-02-15", "valor": 2500 }] }</pre>
7	<p>Enviar na Ferramenta Postman o Método GET para retornar uma receita específica</p> <p>Método: GET</p> <p>API: localhost:8080/api/receitas/1</p> <p>Obs: Neste caso estamos listando a receita com o id=1</p>	<p>sistema deverá validar o JSON e Retorna apenas as informações da receita solicitada</p> <pre>{ "id": 1, "pessoald": 1, "data": "2021-01-12", "valor": 1100.99 }</pre>

8	<p>Enviar na Ferramenta Postman o Método GET para retornar uma receita específica inexistente</p> <p>Método: GET</p> <p>API: localhost:8080/api/receitas/15</p> <p>Obs: Neste caso estamos listando a receita com o id=15</p>	<p>sistema deverá validar o JSON e Retorna reponse:</p> <pre>{ "codigoErro": 1, "mensagemErro": "O registro 15 não foi encontrado" }</pre>
9	<p>Enviar na Ferramenta Postman o Método DEL para excluir uma receita</p> <p>Método: DEL</p> <p>API: localhost:8080/api/receitas/1</p> <p>Obs: Neste caso estamos excluindo a receita com o id=1</p>	<p>sistema deverá validar o JSON e excluir a receita solicitada</p>

Cenário 3 - Totalizadores

Objetivo do Teste	1. Calcula o valor total que foi guardado pelas pessoas ativas na viagem	
Critério de Aceite	1. Valor apresentado conforme solicitado	
Pré-condições	1. Postman configurado 2. API configurada	
Passo	Ação do Passo	Resultados Esperados
1	<p>Enviar na Ferramenta Postman o JSON com o Método GET para calcular o valor total do que foi guardado pelas pessoas ativas na viagem</p> <p>Método: GET</p> <p>API: localhost:8080/api/totalizadores</p> <p>Exemplo de valores guardados</p> <ul style="list-style-type: none"> • Se João está ativo na viagem e fez dois depósitos de R\$ 100.00; • E se Maria não está ativa na viagem e fez um depósito de R\$ 50.00; • Então esta API deve retornar o valor 200 (a soma dos dois depósitos de João) 	<p>sistema retorna o response</p> <ul style="list-style-type: none"> • 2500.25

