

Objetivos:

- Evidenciar os testes de API realizados manualmente;
- Automatizar os testes realizados na etapa manual.

Etapa Manual

Método POST - Cadastro da Pessoa

Neste método conforme requisitado foi atribuído a requisição cinco campos diferentes, sendo eles: **nome** (chave principal), **email**, **cpf**, **dataNascimento** e **telefone**. O método POST foi estruturado da seguinte forma:

- URL: <https://learningprime.com.br/api/pessoa>;
- Header: Content Type: Application/Json;

O curl utilizado pelo método no Postman foi:

```
curl --location 'https://learningprime.com.br/api/pessoa' \
--header 'Content-Type: application/json' \
--data-raw '{
  "nome": "Lucas Capucho de Araujo",
  "email": "lucascapucho@hotmail.com",
  "cpf": 42681196809,
  "dataNascimento": "21/06/1996",
  "telefone": "15981672737"
}'
```

Para validar o método foi escolhido testar um caso positivo e outro negativo, sendo o positivo com o BDD:

```
@post @positivo @200
Cenário: Cadastrar uma Pessoa
  Dado o endereço correto da API
  Quando realizar uma requisição para cadastrar uma Pessoa
  Então a API irá retornar os dados do cadastro da Pessoa respondendo o código 200
  E realizar uma requisição passando o Nome da Pessoa para validar a criação
```

E o negativo com o BDD:

```
@post @erro @400
Cenário: Cadastrar uma Pessoa sem passar o Nome
  Dado o endereço correto da API
  Quando realizar uma requisição para cadastrar uma Pessoa sem passar o Nome
  Então a API irá retornar os dados do cadastro da Pessoa respondendo o código 400
```

Seu resultado com Status 200 demonstrou que o cadastro foi realizado com sucesso da forma manual e da automatizada.

The screenshot shows a REST client interface with a POST request to `https://learningprime.com.br/api/pessoa`. The request body is a JSON object with the following fields: `nome`, `email`, `cpf`, `dataNascimento`, and `telefone`. The status bar at the bottom indicates a successful response with status 200 OK, a time of 241 ms, and a size of 408 B. Below the response, there is a section for test results and a scenario description.

```
1 {
2   "nome": "Lucas Capucho de Araujo",
3   "email": "lucascapucho@hotmail.com",
4   "cpf": 42681196809,
5   "dataNascimento": "21/06/1996",
6   "telefone": "15981672737"
7 }
```

Body Cookies Headers (7) Test Results Status: 200 OK Time: 241 ms Size: 408 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "nome": "Lucas Capucho de Araujo",
3   "email": "lucascapucho@hotmail.com",
4   "cpf": 42681196809,
5   "dataNascimento": "21/06/1996",
6   "telefone": "15981672737"
7 }
```

@post @positivo @200
Cenário: Cadastrar uma Pessoa # features/specifications/pessoa.feature:10
Dado o endereço correto da API # features/step_definitions/acao.rb:1
Quando realizar uma requisição para cadastrar uma Pessoa # features/step_definitions/post.rb:1
Então a API irá retornar os dados do cadastro da Pessoa respondendo o código 200 # features/step_definitions/post.rb:10
Método Put - Status Code 200
Mensagem da Requisição - OK
Resultado dos testes -
E realizar uma requisição passando o Nome da Pessoa para validar a criação # features/step_definitions/get.rb:5
Resultado dos testes - {"email"=>"lucascapucho@hotmail.com", "cpf"=>42681196809, "dataNascimento"=>"21/06/1996", "telefone"=>"15981672737", "nome"=>"Lucas Capucho de Araujo"}

Já seu resultado com Status 400, demonstrou que tentar cadastrar uma Pessoa sem a chave principal **nome** é impossível.

The screenshot shows a REST client interface with a POST request to `https://learningprime.com.br/api/pessoa`. The request body is a JSON object with the following fields: `email`, `cpf`, `dataNascimento`, and `telefone`. The status bar at the bottom indicates a failed response with status 400 Bad Request, a time of 234 ms, and a size of 359 B. Below the response, there is a section for test results and a scenario description.

```
1 {
2   "email": "lucascapucho@hotmail.com",
3   "cpf": 42681196809,
4   "dataNascimento": "21/06/1996",
5   "telefone": "15981672737"
6 }
```

Body Cookies Headers (7) Test Results Status: 400 Bad Request Time: 234 ms Size: 359 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "data": "É obrigatório informar pelo menos um nome para criar um cadastro!\nJson de Exemplo: {'nome':'primeiro_nome'}"
3 }
```

```

@post @erro @400
Cenário: Cadastrar uma Pessoa sem passar o Nome                                # features/specifications/pessoa.feature:16
Dado o endereço correto da API                                                  # features/step_definitions/acao.rb:1
Quando realizar uma requisição para cadastrar uma Pessoa sem passar o Nome      # features/step_definitions/post.rb:6
Então a API irá retornar os dados do cadastro da Pessoa respondendo o código 400 # features/step_definitions/post.rb:10
Método Put - Status Code 400
Mensagem da Requisição - Bad Request
Resultado dos testes - É obrigatório informar pelo menos um nome para criar um cadastro!
Json de Exemplo: {'nome':'primeiro_nome'}

```

Método GET - Consultar dados da Pessoa Cadastrada

Neste método foi passado o atributo **nome** (chave principal) para realizar a consulta da pessoa cadastrada. O método GET foi estruturado da seguinte forma:

- URL: <https://learningprime.com.br/api/nome>;
- Header: Content Type: Application/Json;

O curl utilizado pelo método no Postman foi:

```

curl --location
'https://learningprime.com.br/api/nome/Lucas%20Capucho%20de%20Araujo' \
--header 'Content-Type: application/json'

```

Para validar o método foi escolhido testar um caso positivo e outro negativo, sendo o positivo com o BDD:

```

@get @positivo @200
Cenário: Consultar uma Pessoa
Dado o endereço correto da API
Quando realizar uma requisição passando o Nome da Pessoa
Então a API irá retornar os dados da Pessoa correspondente
respondendo o código 200

```

E o negativo com o BDD:

```

@get @erro @404
Cenário: Consultar uma Pessoa sem passar o Nome
Dado o endereço correto da API
Quando realizar uma requisição sem passar o Nome da Pessoa
Então a API irá retornar os dados da Pessoa correspondente
respondendo o código 404

```

Seu resultado com Status 200 demonstrou que foi possível obter os dados da Pessoa cadastrada anteriormente de forma correta tanto da forma manual como da forma automatizada.

GET ▼ <https://learningprime.com.br/api/nome/Lucas%20Capucho%20de%20Araujo>

Params Authorization **Headers (7)** Body Pre-request Script Tests Settings

Headers 6 hidden

| | Key | Value | Description |
|-------------------------------------|--------------|------------------|-------------|
| <input checked="" type="checkbox"/> | Content-Type | application/json | |
| | Key | Value | Description |

Body Cookies Headers (7) Test Results 🔗 Status: 200 OK Time: 233 ms Size: 380 B

Pretty Raw Preview Visualize JSON ▼ ⇒

```
1  {
2    "data": {
3      "nome": "Lucas Capucho de Araujo",
4      "email": "lucascapucho@hotmail.com",
5      "cpf": 42681196809,
6      "dataNascimento": "21/06/1996",
7      "telefone": "15981672737"
8    }
9  }
```

@get @positivo @200
Cenário: Consultar uma Pessoa # features/specifications/pessoa.feature:22
Dado o endereço correto da API # features/step_definitions/acesso.rb:1
Quando realizar uma requisição passando o Nome da Pessoa # features/step_definitions/get.rb:1
Então a API irá retornar os dados da Pessoa correspondente respondendo o código 200 # features/step_definitions/get.rb:9
Método Put - Status Code 200
Mensagem da Requisição - OK
Resultado dos testes - {"email"=>"lucascapucho@hotmail.com", "cpf"=>42681196809, "dataNascimento"=>"21/06/1996", "telefone"=>"15981672737", "nome"=>"Lucas Capucho de Araujo"}

Seu resultado com Status 404 demonstrou que tentar obter dados de uma Pessoa sem a chave principal **nome** não encontra nada.

GET ▼ <https://learningprime.com.br/api/nome/>

Params Authorization **Headers (7)** Body Pre-request Script Tests Settings

Headers 6 hidden

| | Key | Value | Description |
|-------------------------------------|--------------|------------------|-------------|
| <input checked="" type="checkbox"/> | Content-Type | application/json | |
| | Key | Value | Description |

Body Cookies Headers (8) Test Results 🔗 Status: 404 Not Found Time: 221 ms Size: 278 B

Pretty Raw Preview Visualize JSON ▼ ⇒

```
1  {
2    "data": "Nao Encontrado!"
3  }
```

@get @erro @404
Cenário: Consultar uma Pessoa sem passar o Nome # features/specifications/pessoa.feature:28
Dado o endereço correto da API # features/step_definitions/acesso.rb:1
Quando realizar uma requisição sem passar o Nome da Pessoa # features/step_definitions/get.rb:5
Então a API irá retornar os dados da Pessoa correspondente respondendo o código 404 # features/step_definitions/get.rb:9
Método Put - Status Code 404
Mensagem da Requisição - Not Found
Resultado dos testes - Nao Encontrado!

Método PUT - Edição de uma Pessoa Cadastrada

Neste método foi passado como parâmetro o campo **nome (chave principal)** e o campo **telefone (campo alterado)**. O método PUT foi estruturado da seguinte forma:

- URL: <https://learningprime.com.br/api/nome>;
- Header: Content Type: Application/Json;

O curl utilizado pelo método no Postman foi:

```
curl --location --request PUT
'https://learningprime.com.br/api/nome/Lucas%20Capucho%20de%20Araujo' \
--header 'Content-Type: application/json' \
--data '{
  "telefone": "(15)981672737"
}'
```

Para validar o método foi escolhido testar um caso positivo e outro negativo, sendo o positivo com o BDD:

```
@put @positivo @200
Cenário: Alterar informações de uma Pessoa
  Dado o endereço correto da API
  Quando realizar uma requisição para alterar uma Pessoa
  Então a API irá retornar os dados da Pessoa alterados respondendo o
código 200
```

E o negativo com o BDD:

```
@put @erro @404
Cenário: Alterar informações de uma Pessoa não cadastrada
  Dado o endereço correto da API
  Quando realizar uma requisição para alterar uma Pessoa sem passar o
Nome
  Então a API irá retornar os dados da Pessoa alterados respondendo o
código 404
```

Seu resultado com Status 200 demonstrou que a alteração do campo **telefone** foi realizada com sucesso da forma manual e da automatizada.

```
PUT https://learningprime.com.br/api/nome/Lucas%20Capucho%20de%20Araujo

Body
  none form-data x-www-form-urlencoded raw binary GraphQL JSON
  [1] [2] [3]
  1 [2]
  2 {
  3   "telefone": "(15)981672737"
  4 }

Body Cookies Headers (7) Test Results Status: 200 OK Time: 231 ms Size: 382 B
Pretty Raw Preview Visualize JSON
  [1] [2] [3]
  1 [2]
  2 {
  3   "data": {
  4     "nome": "Lucas Capucho de Araujo",
  5     "email": "lucascapucho@hotmail.com",
  6     "cpf": 42681196809,
  7     "dataNascimento": "21/06/1996",
  8     "telefone": "(15)981672737"
  9   }
  10 }

@put @positivo @200
Cenário: Alterar informações de uma Pessoa # features/specifications/pessoa.feature:34
Dado o endereço correto da API # features/step_definitions/acesso.rb:1
Quando realizar uma requisição para alterar uma Pessoa # features/step_definitions/put.rb:1
Então a API irá retornar os dados da Pessoa alterados respondendo o código 200 # features/step_definitions/put.rb:9
Método Put - Status Code 200
Mensagem da Requisição - OK
Resultado dos testes - {"email"=>"lucascapucho@hotmail.com", "cpf"=>42681196809, "dataNascimento"=>"21/06/1996", "telefone"=>"(15)981672737", "nome"=>"Lucas Capucho de Araujo"}
```

Já o resultado com Status 404 demonstrou que tentar alterar dados de uma Pessoa sem a chave principal **nome** retorna apenas um cadastro não encontrado.

```
PUT https://learningprime.com.br/api/nome/

Body
  none form-data x-www-form-urlencoded raw binary GraphQL JSON
  [1] [2] [3]
  1 [2]
  2 {
  3   "telefone": "(15)981672737"
  4 }

Body Cookies Headers (8) Test Results Status: 404 Not Found Time: 240 ms Size: 278 B
Pretty Raw Preview Visualize JSON
  [1] [2] [3]
  1 [2]
  2 {
  3   "data": "Nao Encontrado!"
  4 }

@put @erro @404
Cenário: Alterar informações de uma Pessoa não cadastrada # features/specifications/pessoa.feature:40
Dado o endereço correto da API # features/step_definitions/acesso.rb:1
Quando realizar uma requisição para alterar uma Pessoa sem passar o Nome # features/step_definitions/put.rb:5
Então a API irá retornar os dados da Pessoa alterados respondendo o código 404 # features/step_definitions/put.rb:9
Método Put - Status Code 404
Mensagem da Requisição - Not Found
Resultado dos testes - Nao Encontrado!
```

Método DELETE – Deletar Pessoa Cadastrada

Neste método foi passado o campo **nome (chave principal)** para realizar a consulta da pessoa cadastrada e então deletá-la. O método DELETE foi estruturado da seguinte forma:

- URL: <https://learningprime.com.br/api/nome>;
- Header: Content Type: Application/Json;

O curl utilizado pelo método no Postman foi:

```
curl --location --request DELETE
'https://learningprime.com.br/api/nome/Lucas%20Capucho%20de%20Araujo' \
--header 'Content-Type: application/json'
```

Para validar o método foi escolhido testar um caso positivo e outro negativo, sendo o positivo com o BDD:

```
@delete @positivo @204
Cenário: Deletar uma Pessoa
  Dado o endereço correto da API
  Quando realizar uma requisição para excluir uma Pessoa
  Então a API deverá retornar os dados da exclusão respondendo o código
204
```

E o negativo com o BDD:

```
@delete @erro @404
Cenário: Deletar uma Pessoa sem passar o Nome
  Dado o endereço correto da API
  Quando realizar uma requisição para excluir uma Pessoa sem passar o
Nome
  Então a API deverá retornar os dados da exclusão respondendo o código
404
```


Seu resultado com Status 204 demonstrou que foi possível deletar a Pessoa utilizando a chave principal **Nome** com sucesso da forma manual e da automatizada.


DELETE <https://learningprime.com.br/api/nome/Lucas%20Capucho%20de%20Araujo>

Params Authorization **Headers (7)** Body Pre-request Script Tests Settings

Headers 6 hidden

| | Key | Value | Description |
|-------------------------------------|--------------|------------------|-------------|
| <input checked="" type="checkbox"/> | Content-Type | application/json | |
| | Key | Value | Description |

Body Cookies Headers (5) Test Results  Status: **204 No Content** Time: 233 ms Size: 184 B

Pretty Raw Preview Visualize Text 

```

1

@delete @positivo @204
Cenário: Deletar uma Pessoa # features/specifications/pessoa.feature:47
Dado o endereço correto da API # features/step_definitions/acesso.rb:1
Quando realizar uma requisição para excluir uma Pessoa # features/step_definitions/delete.rb:1
Então a API deverá retornar os dados da exclusão respondendo o código 204 # features/step_definitions/delete.rb:9
Método Delete - Status Code 204
Mensagem da Requisição - No Content
Resultado dos testes -
  
```


Por fim, o resultado com Status 404 demonstrou que tentar apagar uma Pessoa sem a chave principal **nome** retorna apenas um cadastro não encontrado.


DELETE <https://learningprime.com.br/api/nome/>

Params Authorization **Headers (7)** Body Pre-request Script Tests Settings

Headers 6 hidden

| | Key | Value | Description |
|-------------------------------------|--------------|------------------|-------------|
| <input checked="" type="checkbox"/> | Content-Type | application/json | |
| | Key | Value | Description |

Body Cookies Headers (8) Test Results  Status: **404 Not Found** Time: 222 ms Size: 278 B

Pretty Raw Preview Visualize JSON 

```

1  [24]
2  "data": "Nao Encontrado!"
3  [24]

@delete @erro @404
Cenário: Deletar uma Pessoa sem passar o Nome # features/specifications/pessoa.feature:53
Dado o endereço correto da API # features/step_definitions/acesso.rb:1
Quando realizar uma requisição para excluir uma Pessoa sem passar o Nome # features/step_definitions/delete.rb:5
Então a API deverá retornar os dados da exclusão respondendo o código 404 # features/step_definitions/delete.rb:9
Método Delete - Status Code 404
Mensagem da Requisição - Not Found
Resultado dos testes - Nao Encontrado!
  
```