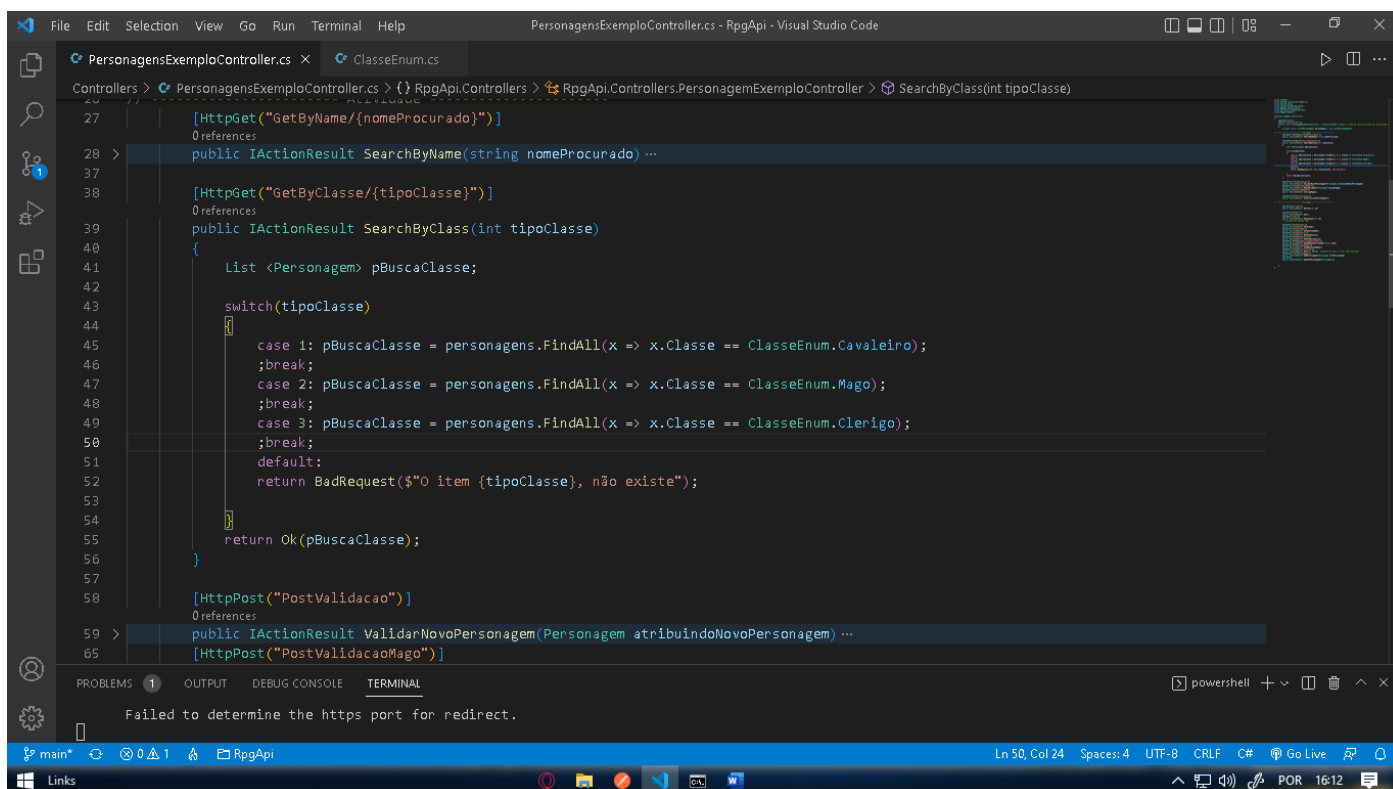


Atividade prática - Controller PersonagensExercicio

1) GetByClasse - Procura pela classe do personagem;

- Código

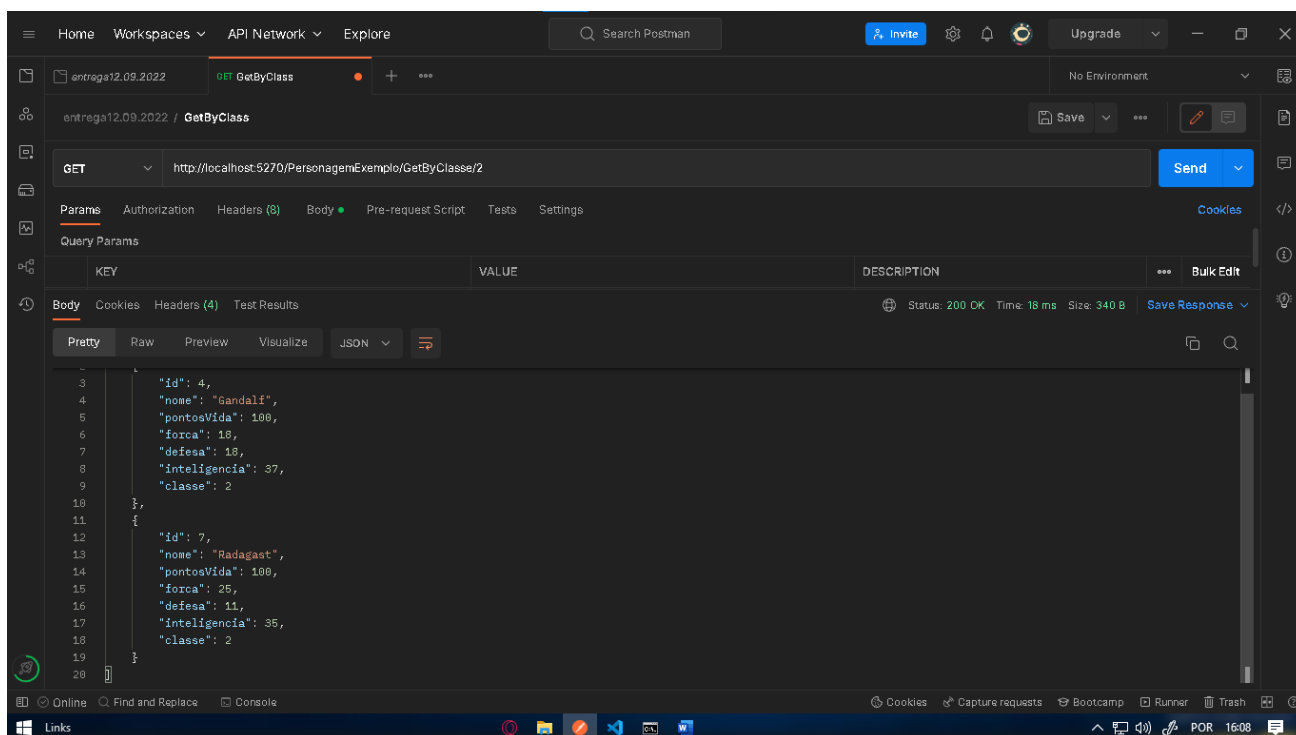


```
27 [HttpGet("GetByName/{nomeProcurado}")]
28 > public IActionResult SearchByName(string nomeProcurado) ...
37
38 [HttpGet("GetByClasse/{tipoClasse}")]
39 0 references
40 public IActionResult SearchByClass(int tipoClasse)
41 {
42     List <Personagem> pBuscaClasse;
43
44     switch(tipoClasse)
45     {
46         case 1: pBuscaClasse = personagens.FindAll(x => x.Classe == ClasseEnum.Cavaleiro);
47         ;break;
48         case 2: pBuscaClasse = personagens.FindAll(x => x.Classe == ClasseEnum.Mago);
49         ;break;
50         case 3: pBuscaClasse = personagens.FindAll(x => x.Classe == ClasseEnum.Clerigo);
51         ;break;
52         default:
53             return BadRequest($"O item {tipoClasse}, não existe");
54     }
55     return Ok(pBuscaClasse);
56 }
57
58 [HttpPost("PostValidacao")]
59 > public IActionResult ValidarNovoPersonagem(Personagem atribuindoNovoPersonagem) ...
65 [HttpPost("PostValidacaoMago")]
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL

Failed to determine the https port for redirect.

- Execução - sem erros



entrega12.09.2022 / **GetByClass**

GET http://localhost:5270/PersonagemExemplo/GetByClasse/2

Params Authorization Headers (3) Body Pre-request Script Tests Settings Cookies

Query Params

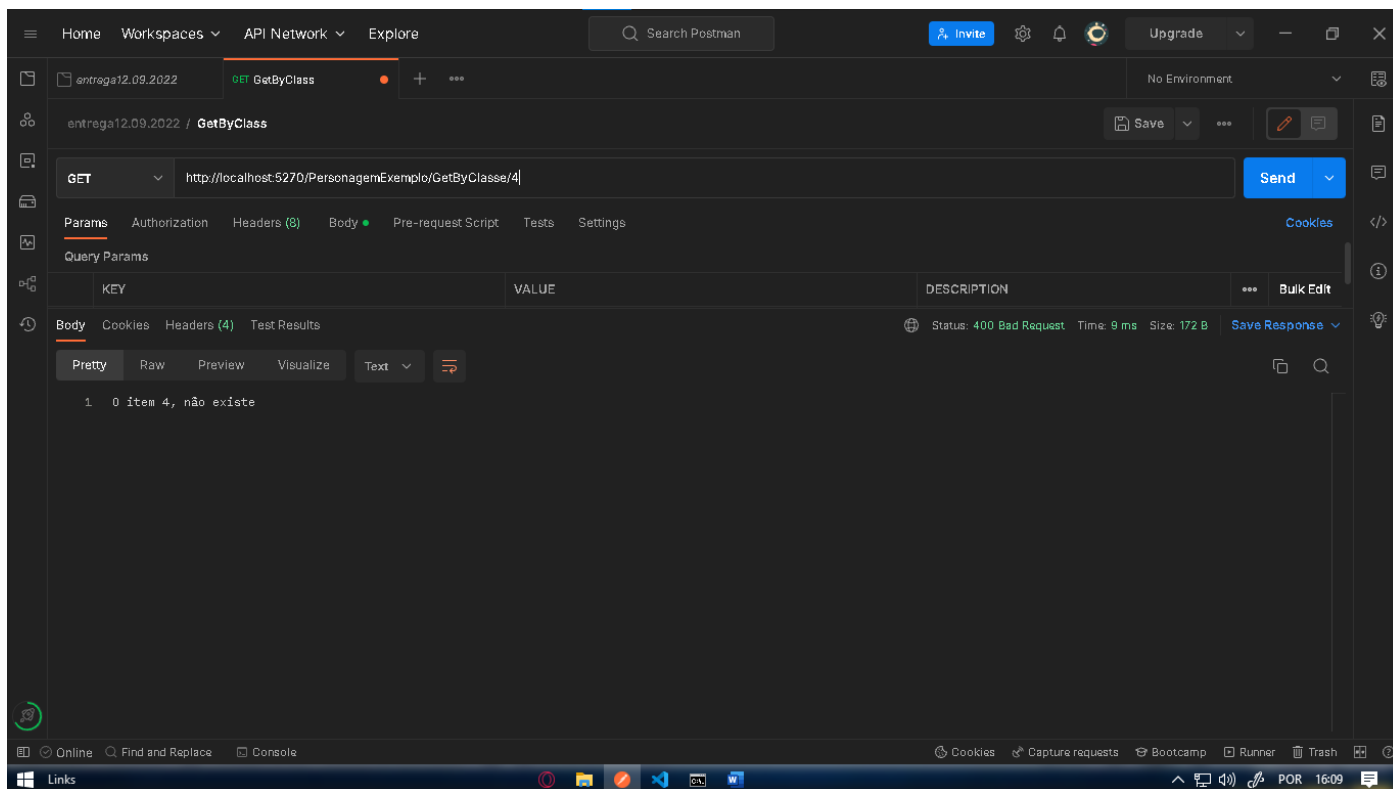
KEY	VALUE	DESCRIPTION
-----	-------	-------------

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 16 ms Size: 340 B Save Response

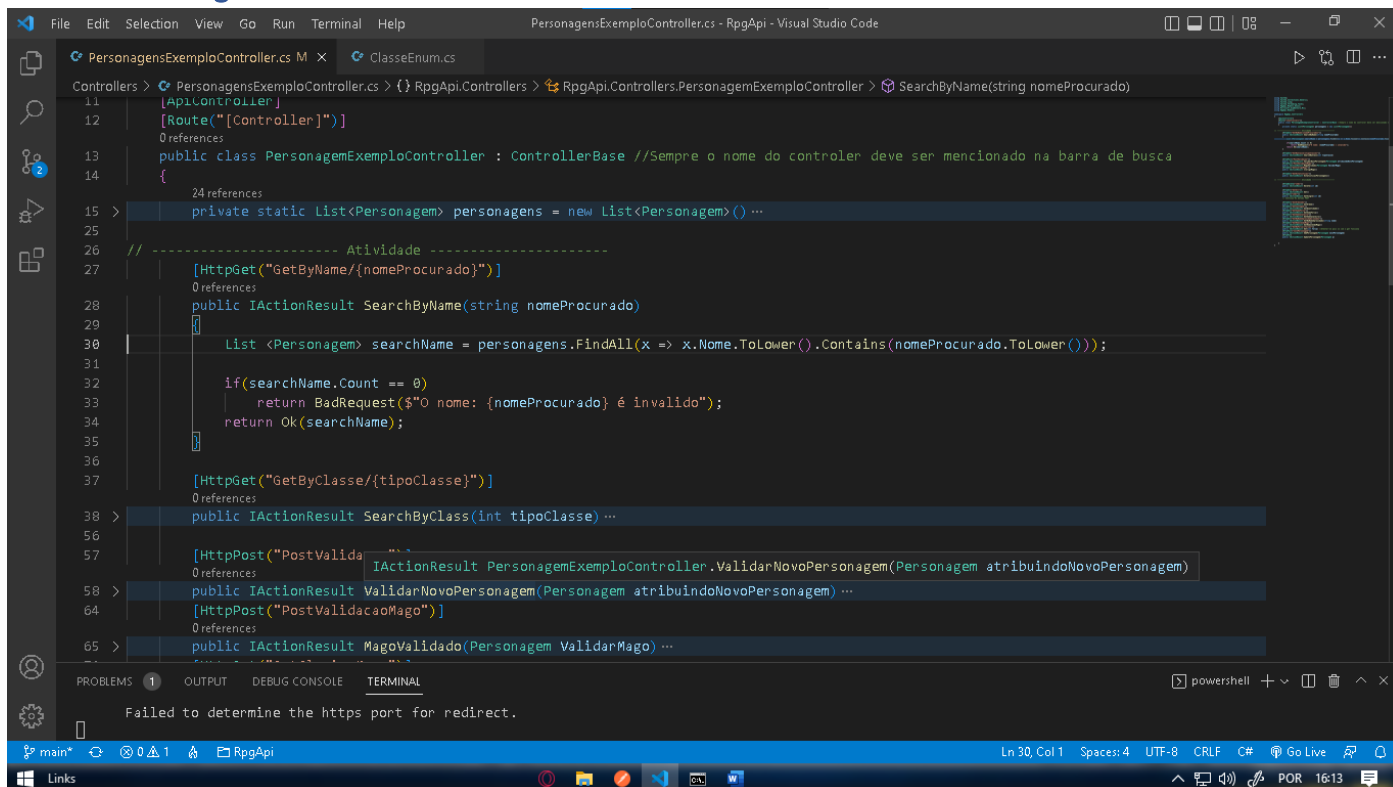
```
3 {
4   "id": 4,
5   "nome": "Gandalf",
6   "pontoesVida": 100,
7   "forca": 10,
8   "defesa": 10,
9   "inteligencia": 37,
10  "classe": 2
11 },
12 {
13   "id": 7,
14   "nome": "Radagast",
15   "pontoesVida": 100,
16   "forca": 25,
17   "defesa": 11,
18   "inteligencia": 35,
19   "classe": 2
20 }
```

- Execução - com erros

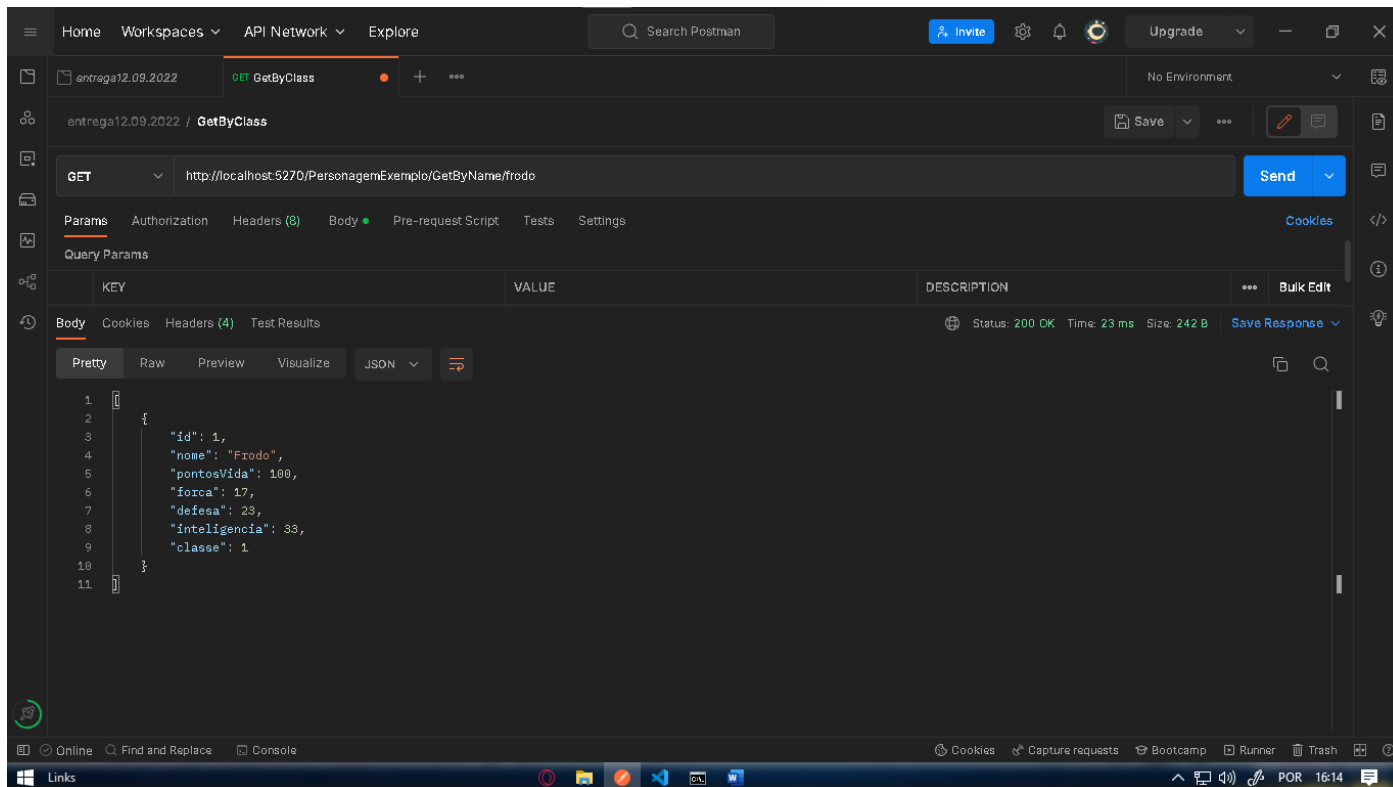


2) GetByNome - procura pelo nome do personagem;

- Código



- Execução - sem erros



entrega12.09.2022 / **GetByClass**

GET `http://localhost:5270/PersonagemExemplo/GetByName/frodo` **Send**

Params Authorization Headers (8) Body • Pre-request Script Tests Settings Cookies

Query Params

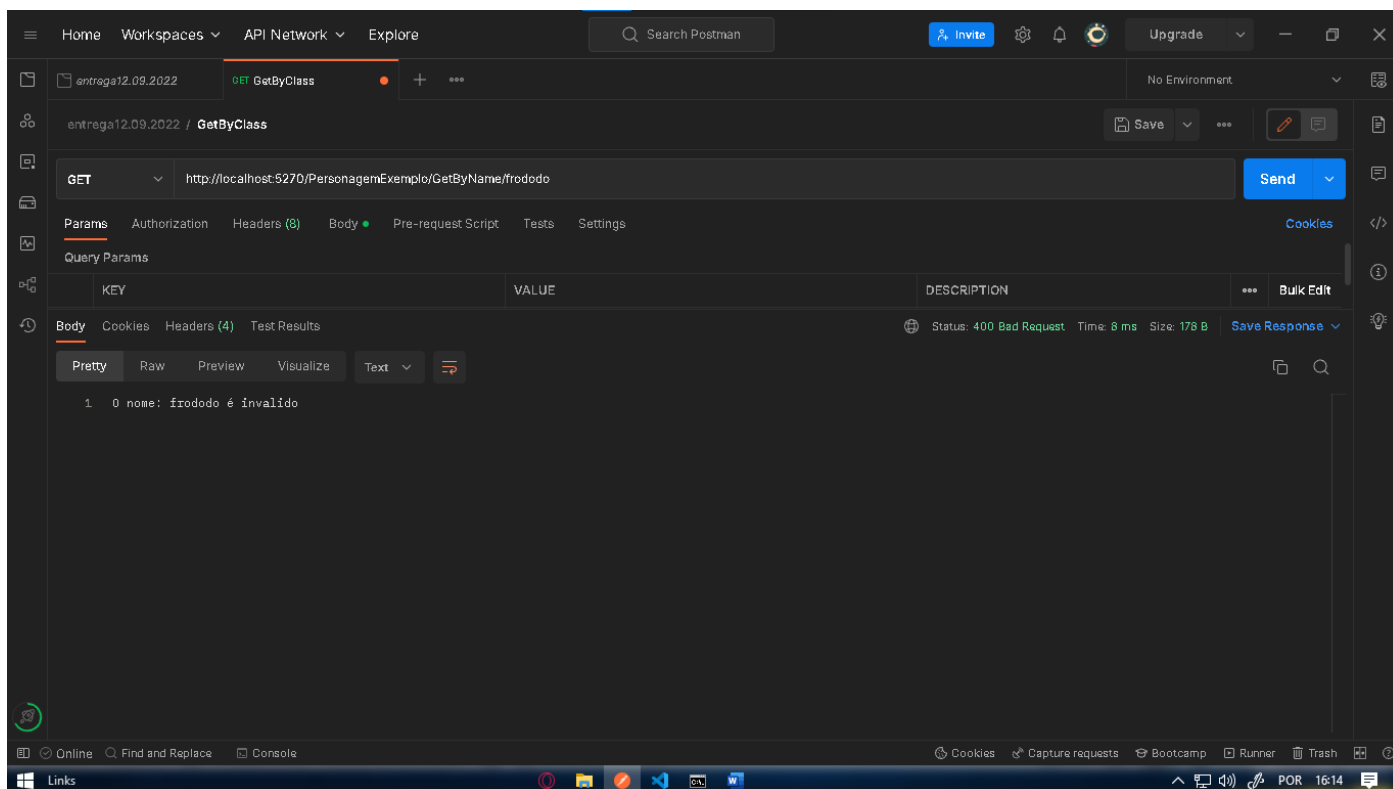
KEY	VALUE	DESCRIPTION	Bulk Edit
-----	-------	-------------	-----------

Body Cookies Headers (4) Test Results Status: 200 OK Time: 23 ms Size: 242 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "id": 1,
3   "nome": "Frodo",
4   "pontosVida": 100,
5   "forca": 17,
6   "defesa": 23,
7   "inteligencia": 33,
8   "classe": 1
9 }
10
11
```

- Execução - com erros



entrega12.09.2022 / **GetByClass**

GET `http://localhost:5270/PersonagemExemplo/GetByName/frodo` **Send**

Params Authorization Headers (8) Body • Pre-request Script Tests Settings Cookies

Query Params

KEY	VALUE	DESCRIPTION	Bulk Edit
-----	-------	-------------	-----------

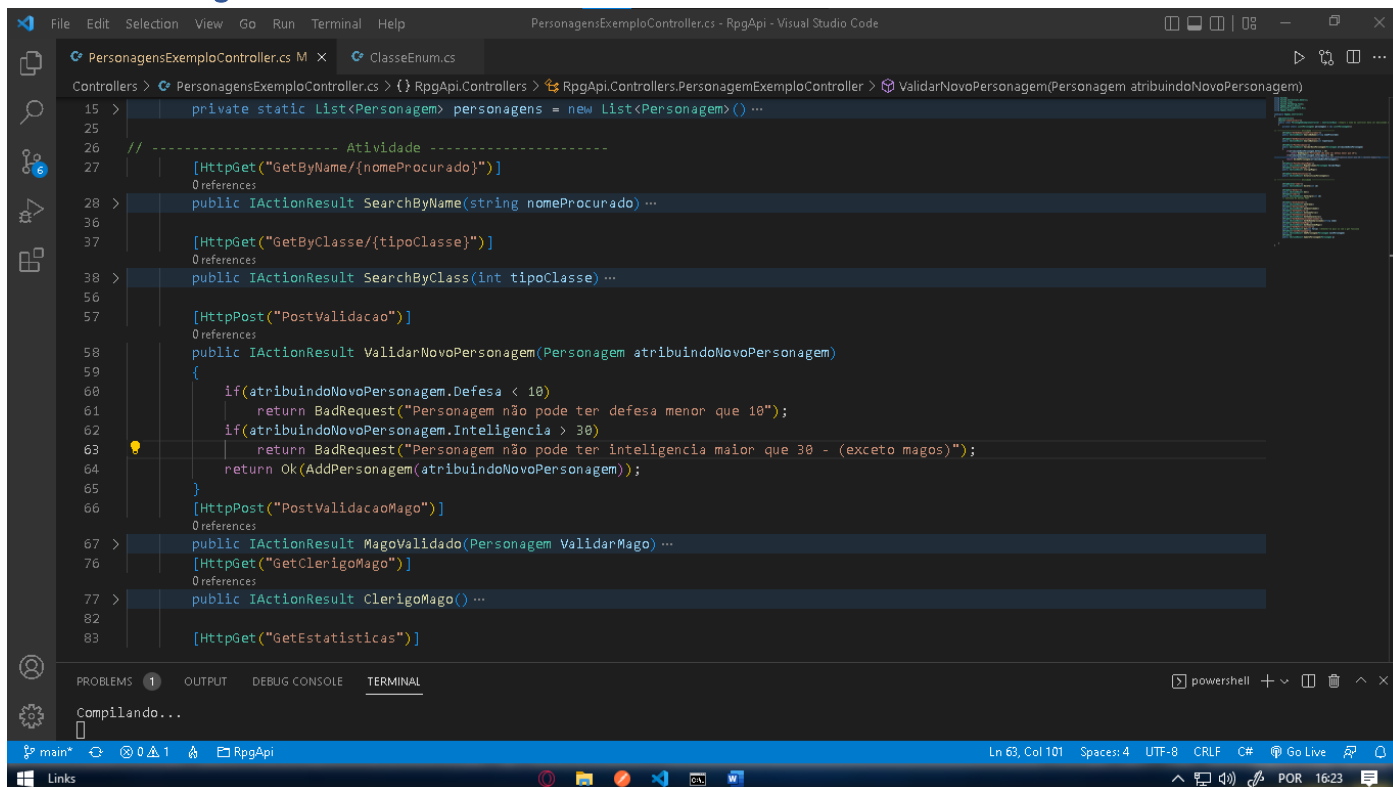
Body Cookies Headers (4) Test Results Status: 400 Bad Request Time: 8 ms Size: 178 B Save Response

Pretty Raw Preview Visualize Text

```
1 0 nome: frodo é inválido
```

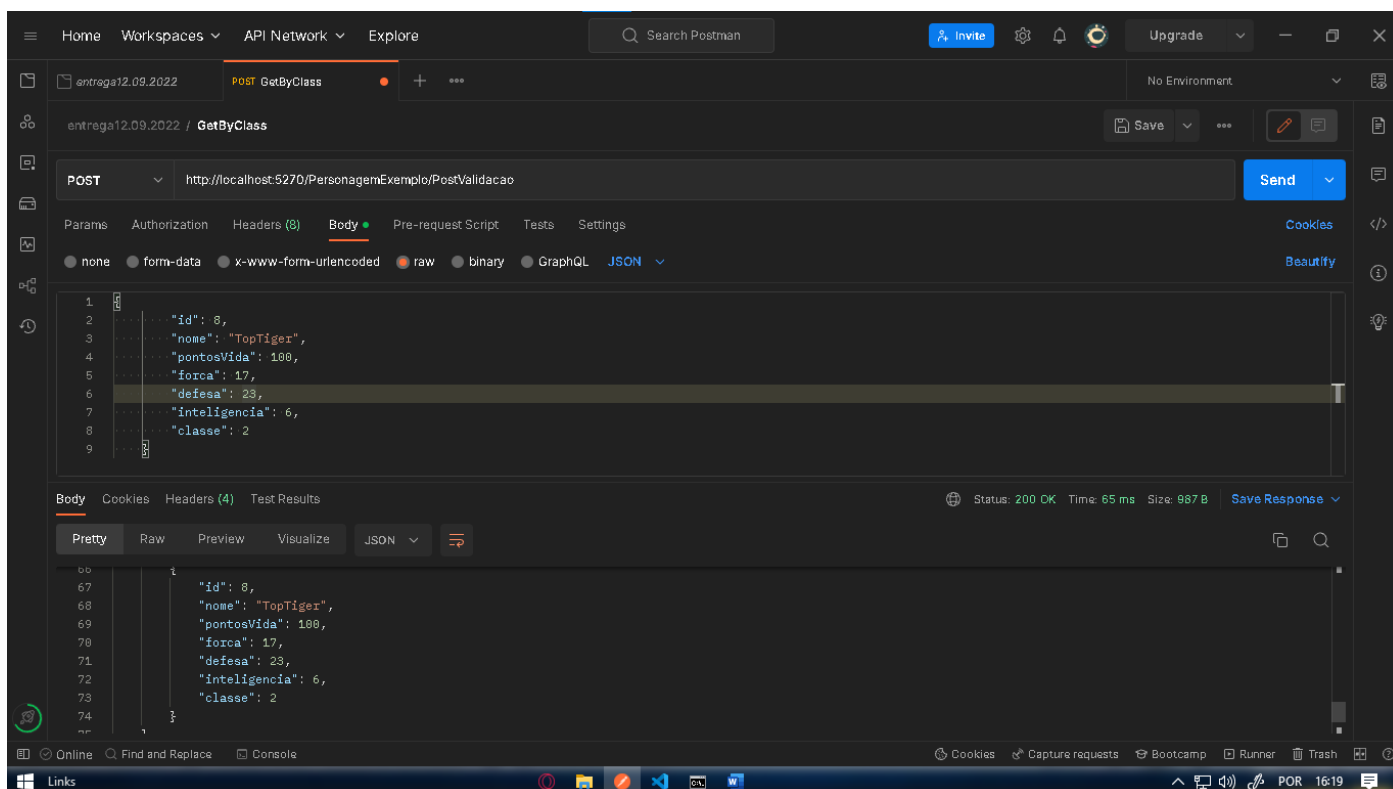
3) PostValidacao - Impede a inclusão do personagem com defesa menor que 10 ou inteligência maior que 30;

- Código



```
15 > private static List<Personagem> personagens = new List<Personagem>() ...
25
26 // ----- Atividade -----
27 [HttpGet("GetByName/{nomeProcurado}")]
28 public IActionResult SearchByName(string nomeProcurado) ...
36
37 [HttpGet("GetByClasse/{tipoClasse}")]
38 public IActionResult SearchByClass(int tipoClasse) ...
56
57 [HttpPost("PostValidacao")]
58 public IActionResult ValidarNovoPersonagem(Personagem atribuindoNovoPersonagem)
59 {
60     if(atribuindoNovoPersonagem.Defesa < 10)
61         return BadRequest("Personagem não pode ter defesa menor que 10");
62     if(atribuindoNovoPersonagem.Inteligencia > 30)
63         return BadRequest("Personagem não pode ter inteligencia maior que 30 - (exceto magos)");
64     return Ok(AddPersonagem(atribuindoNovoPersonagem));
65 }
66 [HttpPost("PostValidacaoMago")]
67 public IActionResult MagoValidado(Personagem ValidarMago) ...
76 [HttpGet("GetClerigoMago")]
77 public IActionResult ClerigoMago() ...
82
83 [HttpGet("GetEstatisticas")]
```

- Execução - sem erros;



entrega12.09.2022 / **GetByClass**

POST http://localhost:5270/PersonagemExemplo/PostValidacao

Params Authorization Headers (8) Body Pre-request Script Tests Settings

Body form-data x-www-form-urlencoded raw binary GraphQL JSON

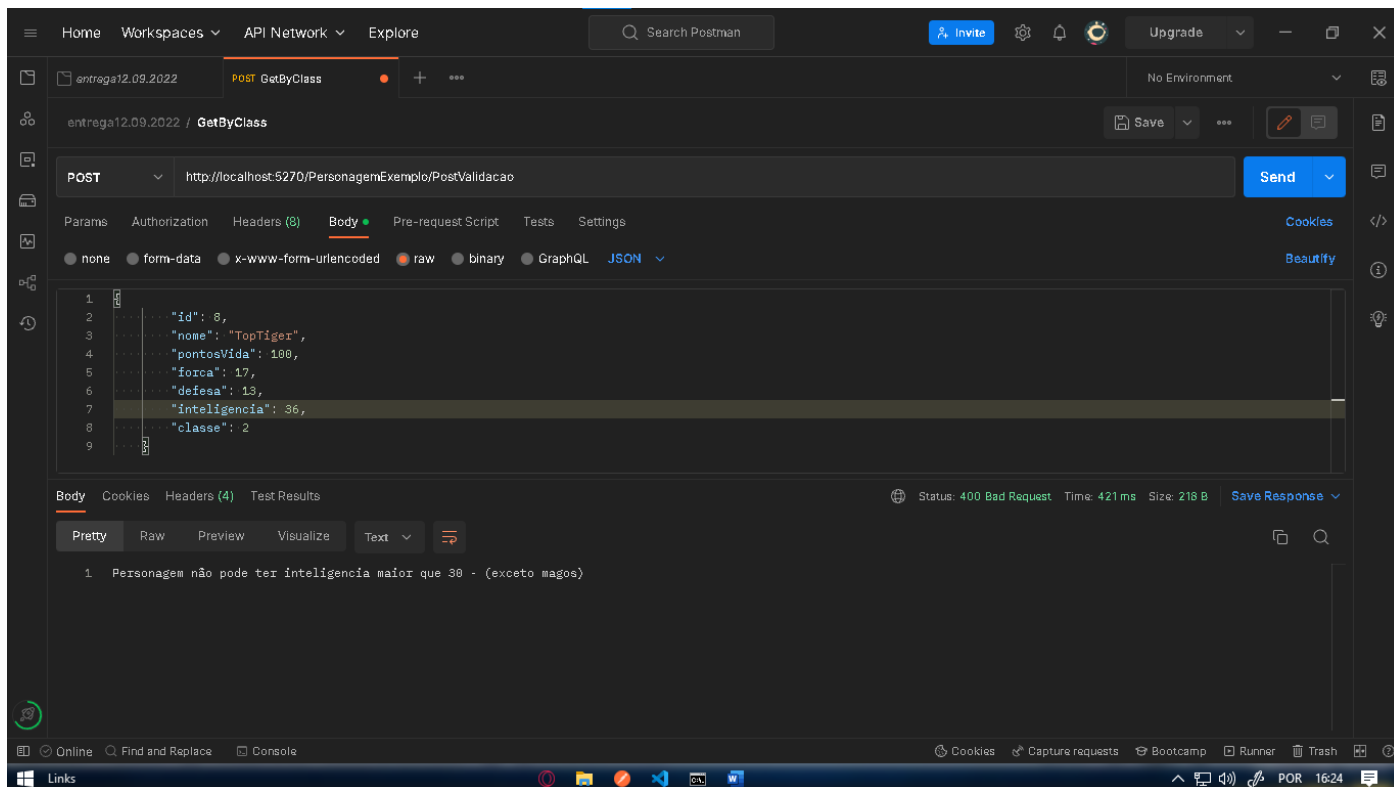
```
1 {
2   "id": 8,
3   "nome": "TopTiger",
4   "pontosVida": 100,
5   "forca": 17,
6   "defesa": 23,
7   "inteligencia": 6,
8   "classe": 2
9 }
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 65 ms Size: 987 B Save Response

```
66 {
67   "id": 8,
68   "nome": "TopTiger",
69   "pontosVida": 100,
70   "forca": 17,
71   "defesa": 23,
72   "inteligencia": 6,
73   "classe": 2
74 }
```

- Execução - com erro na inteligência

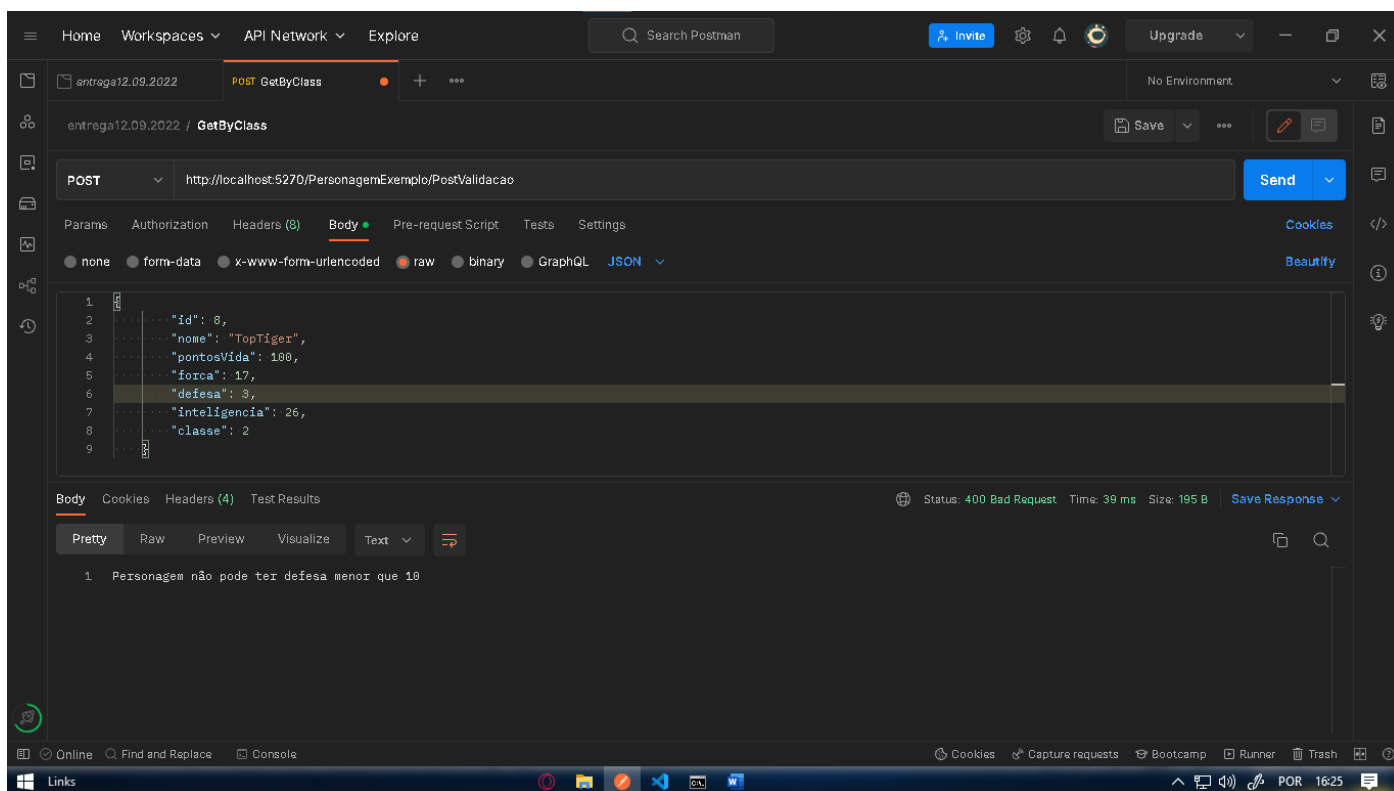


Postman interface showing a POST request to `http://localhost:5270/PersonagemExemplo/PostValidacao`. The request body is a JSON object:

```
{  "id": 8,  "nome": "TopTiger",  "pontosVida": 100,  "forca": 17,  "defesa": 13,  "inteligencia": 36,  "classe": 2}
```

The response status is **400 Bad Request** (Time: 421 ms, Size: 218 B). The error message is: **Personagem não pode ter inteligência maior que 30 - (exceto magos)**.

- Execução - com erro na defesa



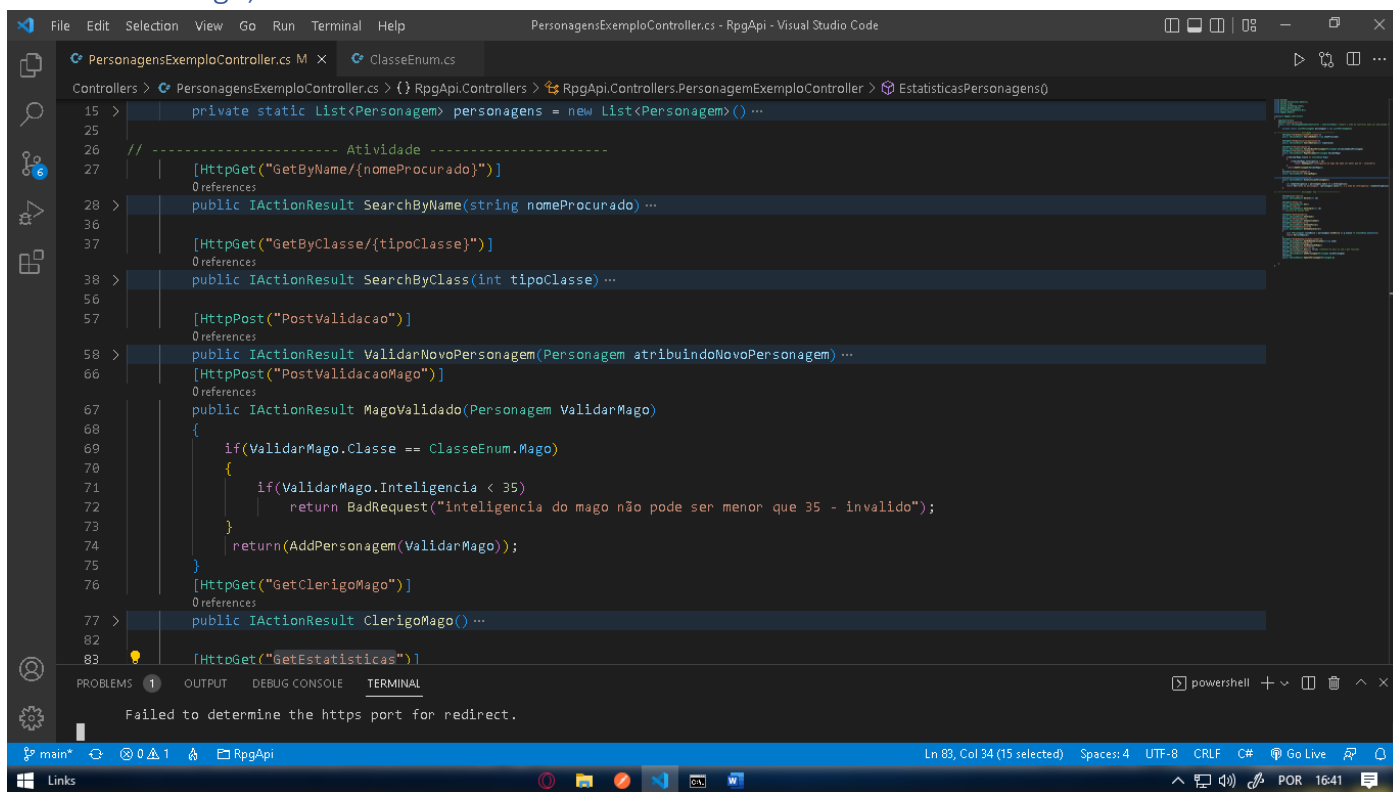
Postman interface showing a POST request to `http://localhost:5270/PersonagemExemplo/PostValidacao`. The request body is a JSON object:

```
{  "id": 8,  "nome": "TopTiger",  "pontosVida": 100,  "forca": 17,  "defesa": 3,  "inteligencia": 26,  "classe": 2}
```

The response status is **400 Bad Request** (Time: 39 ms, Size: 195 B). The error message is: **Personagem não pode ter defesa menor que 10**.

4) PostValidacaoMago - Valida se o mago tem mais de 30 de inteligência, do contrário não entra;

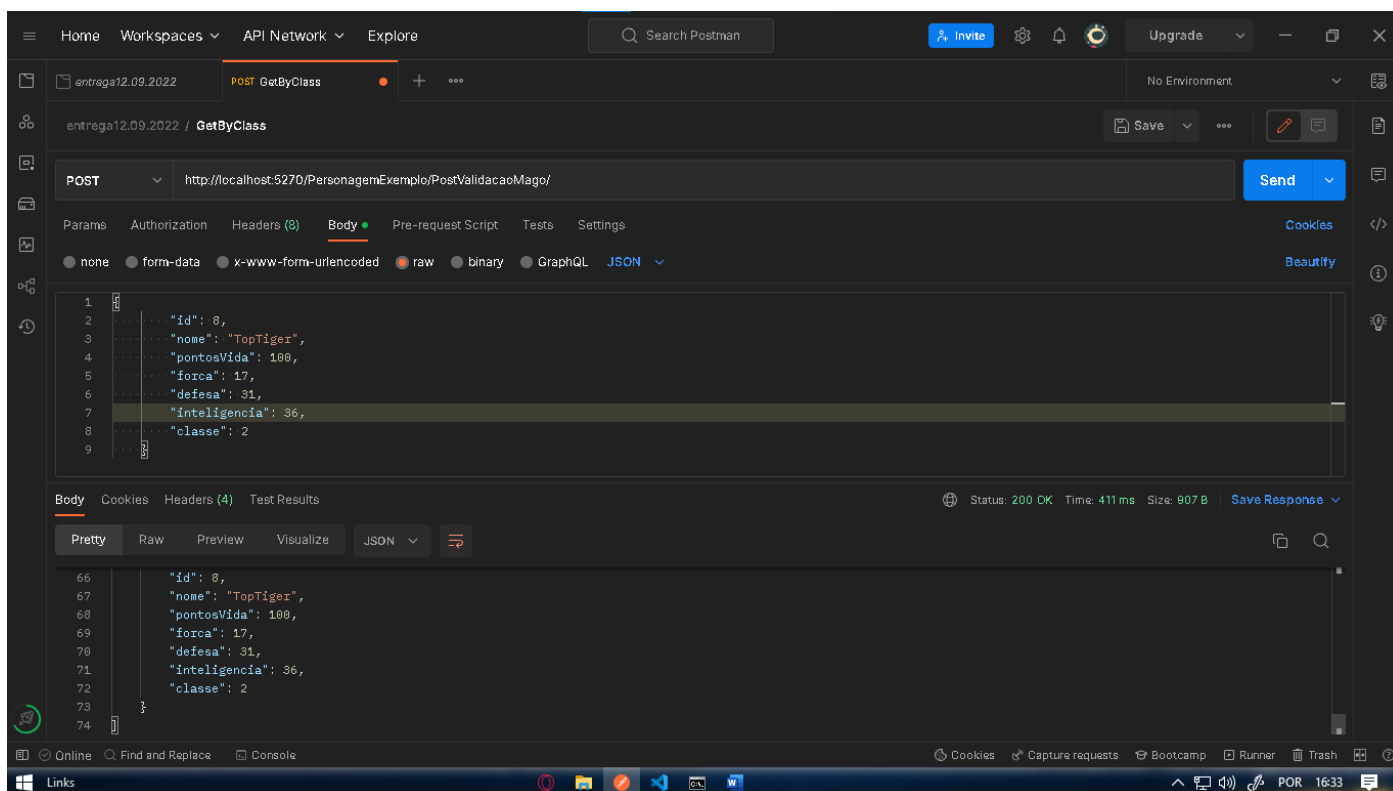
- Código;



```
15 > private static List<Personagem> personagens = new List<Personagem>() ...
25
26 // ----- Atividade -----
27 [HttpGet("GetByName/{nomeProcurado}")]
28 public IActionResult SearchByName(string nomeProcurado) ...
36
37 [HttpGet("GetByClasse/{tipoClasse}")]
38 public IActionResult SearchByClass(int tipoClasse) ...
56
57 [HttpPost("PostValidacao")]
58 public IActionResult ValidarNovoPersonagem(Personagem atribuindoNovoPersonagem) ...
66 [HttpPost("PostValidacaoMago")]
67 public IActionResult MagoValidado(Personagem ValidarMago)
68 {
69     if(ValidarMago.Classe == ClasseEnum.Mago)
70     {
71         if(ValidarMago.Inteligencia < 35)
72             return BadRequest("Inteligencia do mago não pode ser menor que 35 - invalido");
73     }
74     return(AddPersonagem(ValidarMago));
75 }
76 [HttpGet("GetClerigoMago")]
77 public IActionResult ClerigoMago() ...
82
83 [HttpGet("GetEstatisticas")]
```

Failed to determine the https port for redirect.

- Execução - sem erros



entrega12.09.2022 / **GetByClass**

POST http://localhost:5270/PersonagemExemplo/PostValidacaoMago/

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL JSON

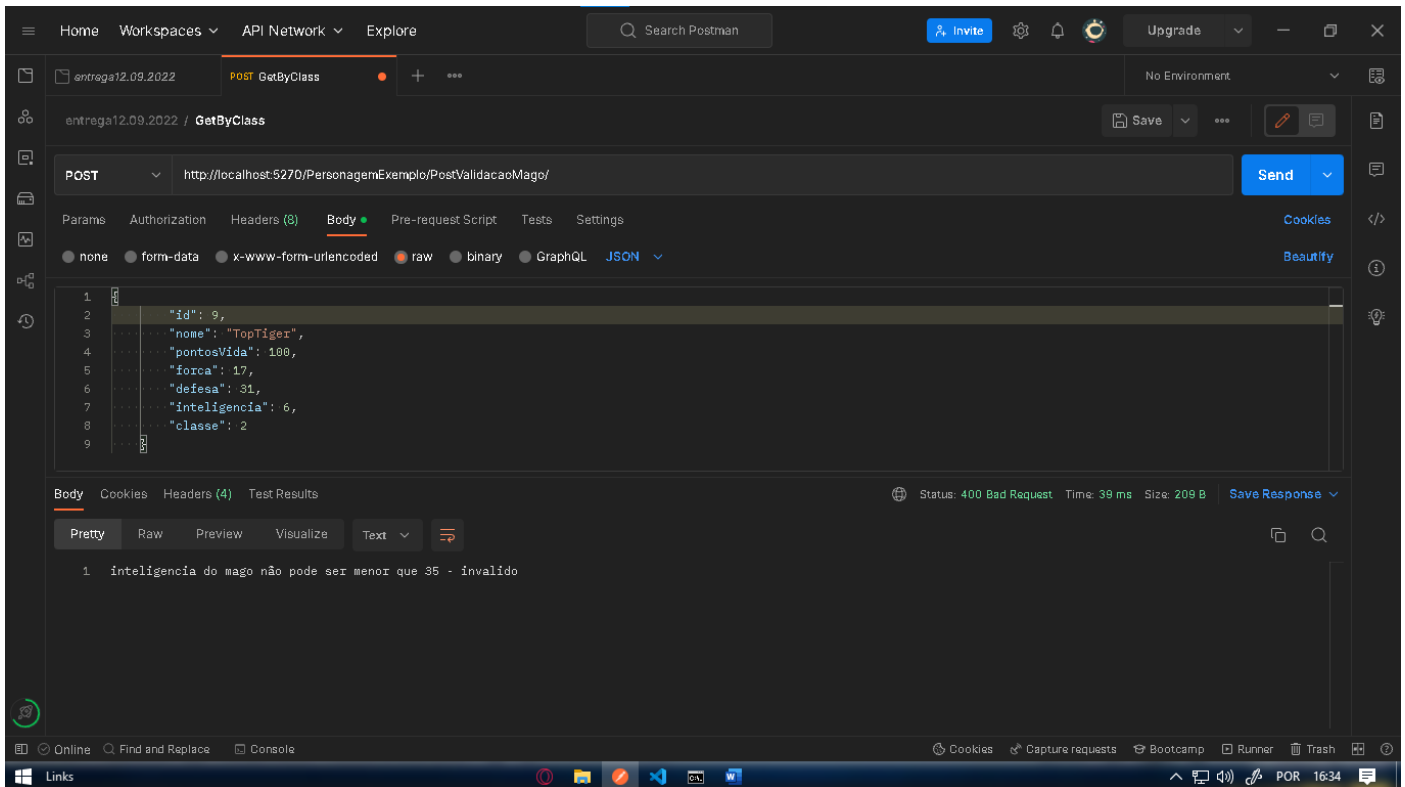
```
1 {
2   "id": 8,
3   "nome": "TopTiger",
4   "pontossVida": 100,
5   "forca": 17,
6   "defesa": 31,
7   "inteligencia": 36,
8   "classe": 2
9 }
```

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 411 ms Size: 907 B Save Response

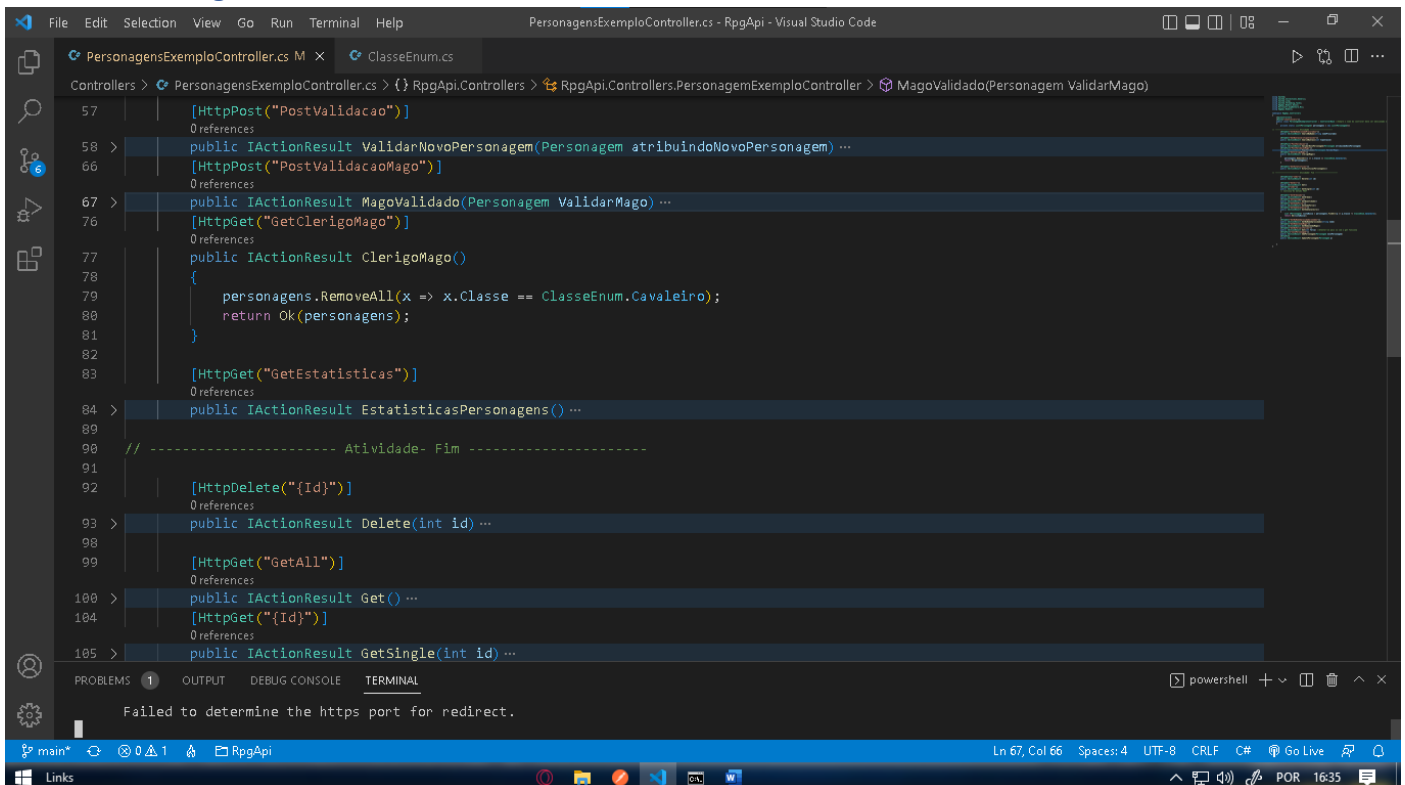
```
66 {
67   "id": 8,
68   "nome": "TopTiger",
69   "pontossVida": 100,
70   "forca": 17,
71   "defesa": 31,
72   "inteligencia": 36,
73   "classe": 2
74 }
```

- Execução - com erros

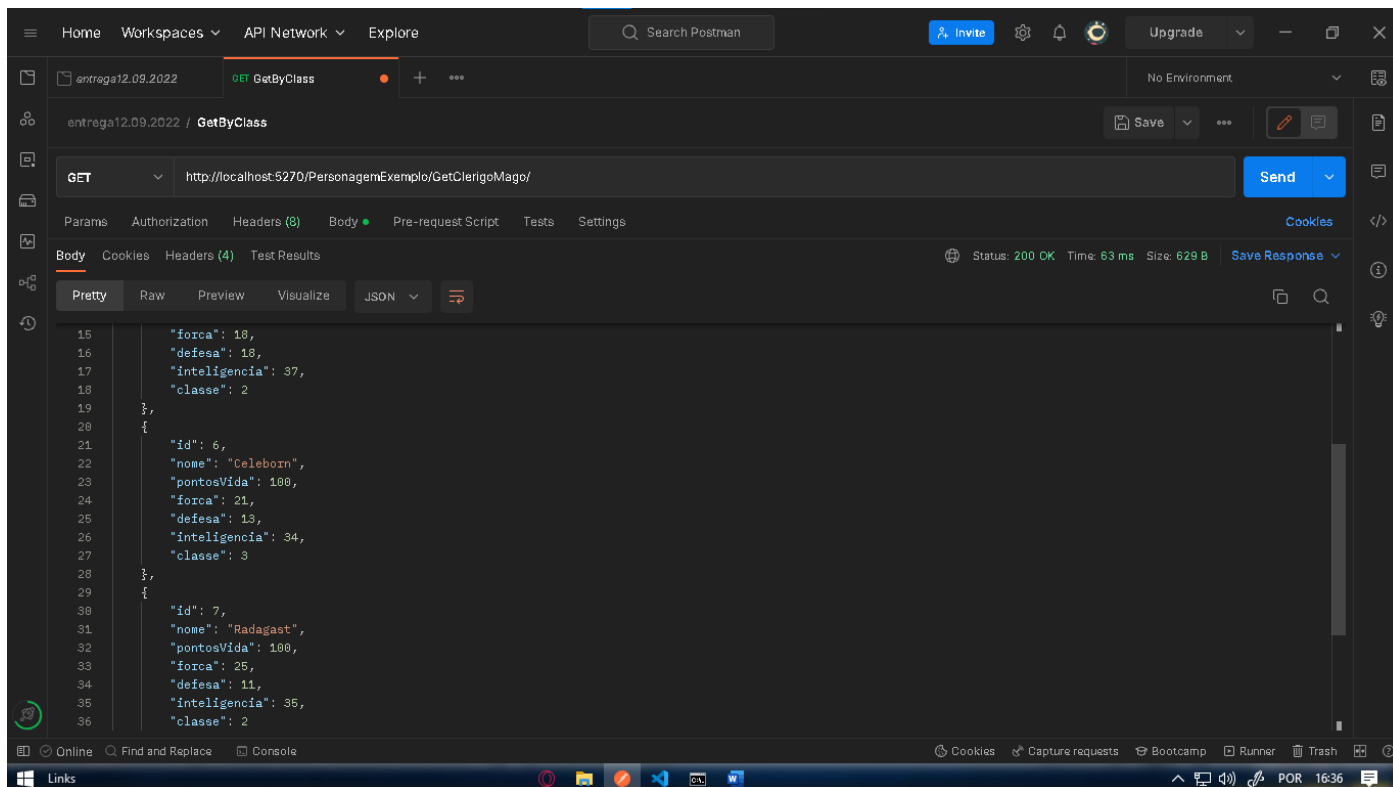


5) GetClerigoMago - exclui todos as classes do tipo cavaleiro.

- Código

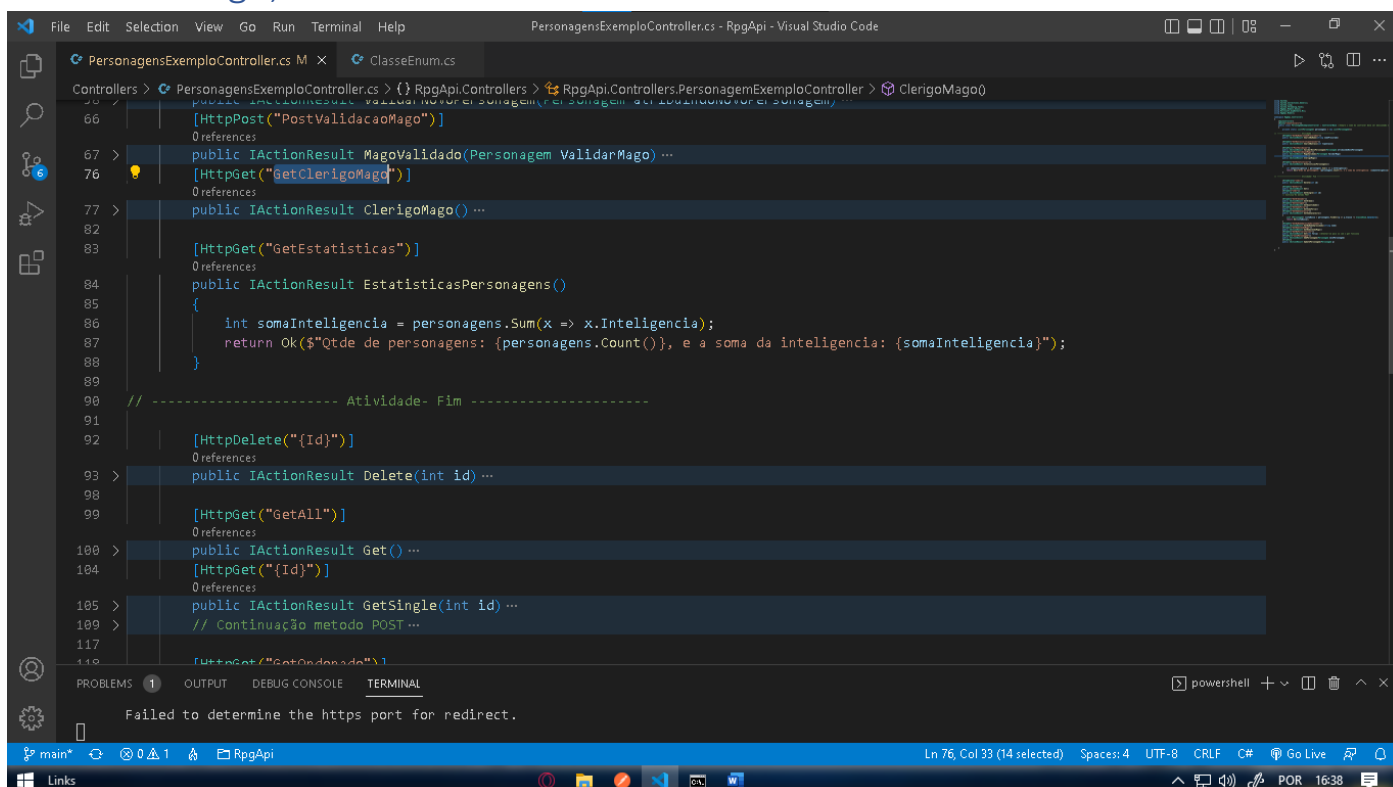


- Execução

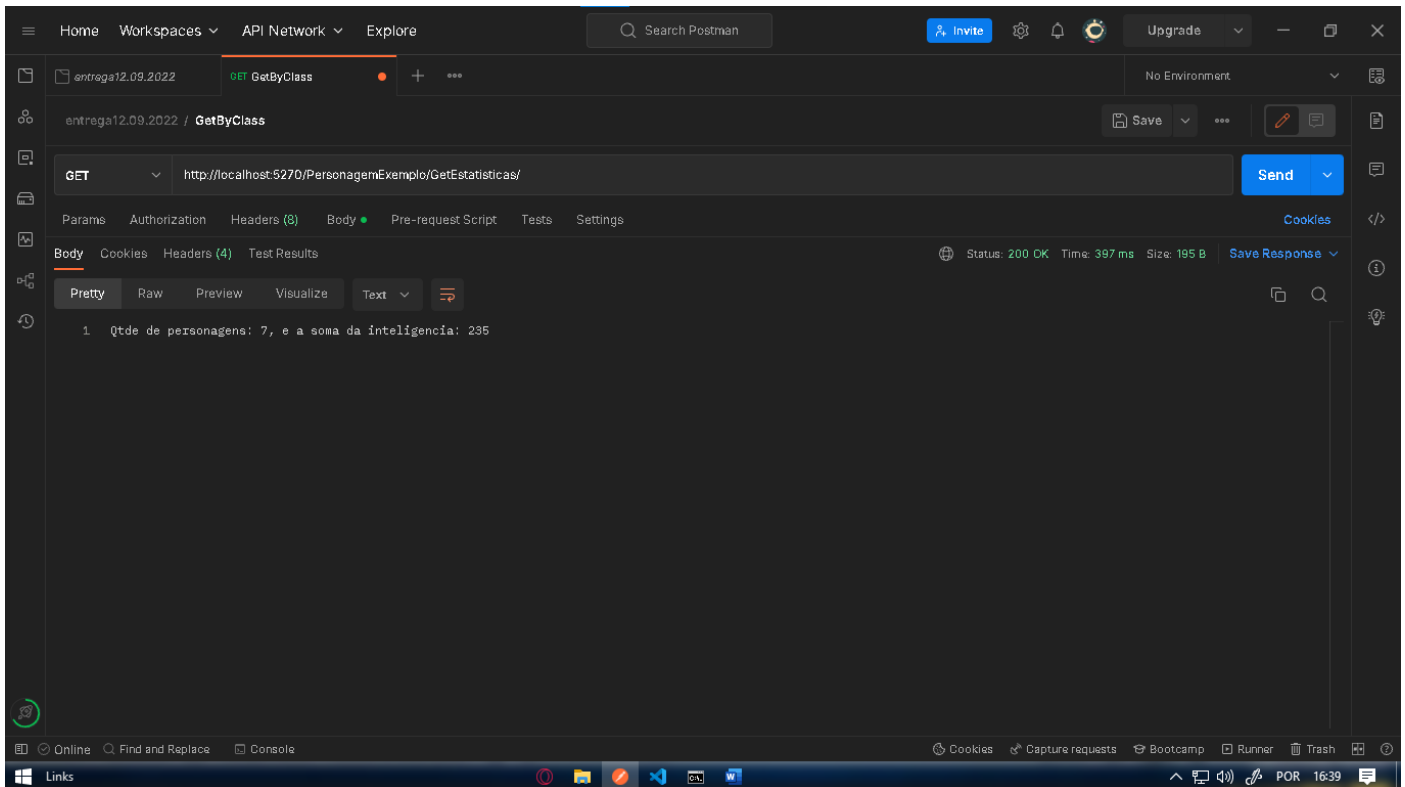


6) GetEstatisticas - mostra a soma total da inteligência e a quantidade de personagens;

- Código;



- Execução



Link GitHub: <https://github.com/DanMarzo/aulas-desenvolvimento-de-sistemas-etec>