



Exercícios para entrega via Forms no Teams

Crie uma controller chamada **PersonagensExercicioController** copie a lista da controller usada em aula, conforme abaixo e construa os seguintes métodos testando a execução no postman:

```
using Microsoft.AspNetCore.Mvc;
using RpgApi.Models;
using RpgApi.Models.Enums;

namespace RpgApi.Controllers
{
    [ApiController]
    [Route("[Controller]")]
    0 references
    public class PersonagensExercicioController : ControllerBase
    {
        0 references
        private static List<Personagem> personagens = new List<Personagem>()
        {
            //Códigos podem ser aproveitados da controller de exemplo
            new Personagem() { Id = 1, Nome = "Frodo", PontosVida=100, Forca=17, Defesa=23, Inteligencia=33, Classe=ClasseEnum.Cavaleiro},
            new Personagem() { Id = 2, Nome = "Sam", PontosVida=100, Forca=15, Defesa=25, Inteligencia=30, Classe=ClasseEnum.Cavaleiro},
            new Personagem() { Id = 3, Nome = "Galadriel", PontosVida=100, Forca=18, Defesa=21, Inteligencia=35, Classe=ClasseEnum.Clerigo },
            new Personagem() { Id = 4, Nome = "Gandalf", PontosVida=100, Forca=18, Defesa=18, Inteligencia=37, Classe=ClasseEnum.Mago },
            new Personagem() { Id = 5, Nome = "Hobbit", PontosVida=100, Forca=20, Defesa=17, Inteligencia=31, Classe=ClasseEnum.Cavaleiro },
            new Personagem() { Id = 6, Nome = "Celeborn", PontosVida=100, Forca=21, Defesa=13, Inteligencia=34, Classe=ClasseEnum.Clerigo },
            new Personagem() { Id = 7, Nome = "Radagast", PontosVida=100, Forca=25, Defesa=11, Inteligencia=35, Classe=ClasseEnum.Mago }
        };

        //Métodos deverão ser construídos aqui
    }
}
```

- Método do tipo *Get* com a rota **GetByNome** que selecione um personagem de acordo com o nome digitado e que caso não encontre retorne a mensagem NotFound informando formalmente na resposta da requisição.
- Método do tipo *Post* com rota **PostValidacao** que não permita que um Personagem seja adicionado tendo o valor de defesa menor que 10 ou inteligência maior que 30. Deve retornar um BadRequest caso esse dado seja enviado na requisição.
- Método do tipo *Post* com rota **PostValidacaoMago** que não permita que um personagem do tipo Mago seja incluído com inteligência menor que 35. Deve retornar um BadRequest caso esse dado seja enviado na requisição.
- Método do tipo *Get* com rota **GetClerigoMago** que remova os personagens que são cavaleiros, exiba a lista decrescente por *PontosVida*.
- Método do tipo *Get* com rota **GetEstatisticas** e exiba a quantidade de personagens da lista e qual o somatório de inteligência dos personagens.
- Método do tipo *Get* com rota **GetByClasse** que selecione a lista de personagens de acordo com o id da classe informada na requisição.