

Atividade 4 — API com Programação Orientada a Objetos em C#/.NET

Este repositório contém a implementação de uma **API REST em C#/.NET** com foco em **Programação Orientada a Objetos (POO)**.

Inclui exemplos de **classe abstrata, herança e polimorfismo**, organizados em controllers, models e services.

Objetivo da Atividade

- Implementar API em C# com separação em Controllers, Models e Services
- Aplicar conceitos de POO (classe abstrata, herança, polimorfismo)
- Consumir a API via Swagger, VS Code REST Client (.http) ou Postman
- Documentar evidências de funcionamento



Estrutura do Projeto

```
/API00Ppersonagens
├── Controllers/
│   └── PersonagensController.cs
├── Models/
│   ├── Abstracoes/
│   │   └── PersonagemBase.cs
│   ├── Ninja.cs
│   ├── Pirata.cs
│   ├── Shinigami.cs
│   └── DTOs/
│       └── CriarPersonagemDto.cs
├── Services/
│   ├── IPersonagemService.cs
│   └── PersonagemService.cs
├── Program.cs
├── API00Ppersonagens.csproj
└── API00Ppersonagens.http (arquivo para VS Code REST Client)
```



Como rodar a API

```
dotnet restore
dotnet build
dotnet run
```

A aplicação deve subir em algo como:

Now listening on: <http://localhost:5275>

Acesse no navegador:

- Swagger: <http://localhost:5275/swagger>
- Endpoint: <http://localhost:5275/personagens>

Endpoints disponíveis

- GET /personagens → lista todos os personagens (com filtro opcional por classe)
- GET /personagens/{id} → busca personagem por ID
- POST /personagens → cria novo personagem (tipo = ninja, pirata, shinigami)
- GET /personagens/{id}/ataque → executa ataque (demonstra polimorfismo)

Exemplo de body JSON para criação:

```
{  
  "tipo": "ninja",  
  "nome": "Kakashi",  
  "universo": "Naruto",  
  "atributo": "Chidori"  
}
```

Testando a API

1. Swagger:

- Acesse <http://localhost:5275/swagger>
- Use Try it out para executar GET e POST

2. VS Code REST Client:

Arquivo API00P.http com requests:

@base = http://localhost:5275

Lista

GET {{base}}/personagens

Buscar por Id

GET {{base}}/personagens/1

Criar novo

POST {{base}}/personagens

Content-Type: application/json

```
{  
  "tipo": "ninja",  
  "nome": "Kakashi",  
  "universo": "Naruto",  
  "atributo": "Chidori"  
}
```



Evidências

As evidências estão documentadas em [Evidencias.md](#).



Conclusão

- API criada com separação em camadas (Controller, Model, Service)
- Conceitos de POO implementados (abstração, herança e polimorfismo)
- Testada via Swagger, VS Code REST Client e Postman
- Evidências anexadas no repositório