

Consolidando o seu conhecimento

Chegou a hora de você seguir todos os passos realizados por mim durante esta aula:

- Instalar e configurar o **PHP** na versão **7.3** ou maior, **Composer** e **MySQL Server**
- Extrair o projeto inicial do treinamento, que pode ser baixado [aqui \(https://caelum-online-public.s3.amazonaws.com/1158-php-symfony-api-restful-autenticacao-relacionamentos/01/1158-api-symfony-parte2-projeto-inicial.zip\)](https://caelum-online-public.s3.amazonaws.com/1158-php-symfony-api-restful-autenticacao-relacionamentos/01/1158-api-symfony-parte2-projeto-inicial.zip).
- Através do console, acesse a pasta do projeto e execute:

```
composer install
```

- Ainda no console, dentro da pasta do projeto, execute as *migrations*:

```
php bin/console doctrine:migrations:migrate
```

- Crie e implemente a classe `MedicosRepository`, baseando-se na classe `EspecialidadeRepository`
- Na classe `Medico`, dentro da anotação `@ORM\Entity()`, informe o caminho do seu repositório
- Na construtor da classe `MedicosController`, injete um `MedicosRepository` e atribua-o a um atributo privado da classe
 - Nos métodos `buscarTodos` e `buscaMedico`, remova o código que busca o médico e passe a utilizar o `MedicosRepository`
- Ainda na classe `MedicosController`, crie e implemente o método `buscaPorEspecialidade`, que recebe `especialidadeId` (inteiro) como parâmetro, mapeia a rota `/especialidade/{especialidadeId}/medicos` e só aceita requisições do tipo GET
 - No repositório de médicos, esse método irá buscar os médicos que tenham a `especialidade` igual à `especialidadeId` recebida por parâmetro, e retorná-los em um `JsonResponse`

Caso já tenha feito, excelente. Se ainda não, é importante que você execute o que foi visto nos vídeos para poder continuar com a próxima aula.