

Consolidando o seu conhecimento

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

- Crie a classe `ExtratorDadosRequest`
 - Implemente o método privado `buscaDadosRequest`, que recebe um `Request` por parâmetro
 - Esse método irá buscar as informações de ordenação da *query string* do `Request`, em seguida buscar todas as informações da *query string* do `Request`, que serão as informações de filtro, e remover dessas informações a chave `sort`, que é a chave da ordenação
 - Por fim, irá retornar um array com as informações de ordenação e as informações de filtro
 - Implemente o método público `buscaDadosOrdenacao`, que recebe um `Request` por parâmetro
 - Esse método irá pegar e retornar somente as informações de ordenação retornadas pelo método `buscaDadosRequest`:

```
[$informacoesDeOrdenacao, ] = $this->buscaDadosRequest($request)
```

- Implemente o método público `buscaDadosFiltro`, que recebe um `Request` por parâmetro
 - Esse método irá pegar e retornar somente as informações de filtro retornadas pelo método `buscaDadosRequest`:

```
[ , $informacoesDeFiltro] = $this->buscaDadosRequest($request)
```

- Injete um `ExtratorDadosRequest` no construtor das classes `MedicosController` e `EspecialidadesController`, passando-o para o construtor da classe mãe
- Injete um `ExtratorDadosRequest` no construtor da classe `BaseController` e atribua-o a um atributo protegido da classe
 - No método `buscarTodos`, busque as informações de ordenação e de filtro através da classe `ExtratorDadosRequest`
 - Em seguida, passe as informações de filtro e informações de ordenação para o método `findBy`, do repositório:

```
$lista = $this->repository->findBy($informacoesDeFiltro, $informacoesDeOrdenacao);
```

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.