



**INSTITUTO FEDERAL**

São Paulo

Câmpus Bragança Paulista

Tecnologia em Análise e Desenvolvimento de Sistemas

Desenvolvimento Web Back-end

Prof. Luiz Gustavo Diniz de Oliveira Vêras

## Melhorias no Projeto StarWars em GraphQL

Nesta atividade, você irá melhorar o código atual do projeto StarWars, aplicando boas práticas de arquitetura em camadas (Controller, Model, Service, Repository) e ampliando a API com novas *queries* e *mutations*.

Para aplicar essas melhorias, inicie baixando o projeto “Aula 4 (Projeto) graphql Spring StarWars” disponível no Moodle e abra-o no Visual Studio Code.

## Refatoração em camadas

Você irá criar as camadas/pacotes Service e Repository. As camadas Controller e Model já devem existir no projeto.

Sobre a camada Service, siga as orientações:

- Mova a lógica que no momento encontra-se em `StarWarController` para classes de serviços. Exemplo: `HeroService`, `DroidService`, `SearchService`, `ReviewService`.

Sobre a camada Repository, siga as orientações

- Crie uma classe de Repository para Humans, Droids e Starships.
- Para cada Repository, crie métodos que representem CRUD para aquela entidade. Por exemplo, para `HumanRepository`, teríamos:

- `List<Human> findAllHumans();`
- `Human findHumanById(String id);`
- `void saveHuman(Human human);`
- `void updateHuman(Human human);`
- `void deleteHuman(String id);`
- Fazer o mesmo acima para `DroidRepository` e `StarshipRepository`.
- Use `ArrayList` como base de dados em memória para Humans, Droids e Starships.

Na classe de *Controller*, agora ela apenas passa a delegar chamadas ao *Service* (comandos do tipo `humanService.findAllHumans()`). Sendo assim, o *controller* deve apenas mapear a *query/mutation* e chamar os serviços.

## Novas Queries

Adicione no schema GraphQL do projeto as novas *queries* além das já existentes:

- Buscar todos os humanos: `humans: [Human!]!`
- Buscar todas as naves: `starships: [Starship!]!`
- Buscar por ID genérico (pode retornar qualquer *Character*): `character(id: ID!): Character`

Para realizar essas implementações, você deve:

- Especificar as queries no schema GraphQL;
- Associá-las à uma função no *Controller*.

## Novas Mutations

Agora adicione as operações de escrita de novos dados, de acordo com as descrições de operações a seguir:

- Criar um novo Human: `createHuman(id: ID!, name: String!, height: Float): Human`
- Criar um novo Droid: `createDroid(id: ID!, name: String!, primaryFunction: String): Droid`

- Criar uma nova Starship: createStarship(id: ID!, name: String!, length: Float): Starship
- Adicionar amigo a um personagem: addFriend(characterId: ID!, friendId: ID!): Character

Para realizar essas implementações, você deve:

- Especificar as mutations no schema GraphQL;
- Associá-las à um método na classe de Controller.