

CENTRO UNIVERSITÁRIO UNIRUY WYDEN
CAMPUS IMBUÍ – SALVADOR BA

ARIANE OLIVEIRA DE SOUZA -RA: 2019 51445384

BRUNO MENEZES DOS SANTOS -RA :2022 0905 3991

**APLICAÇÃO GERENCIAMENTO DE ALUNOS EM UMA
ACADEMIA -WAYFIT**

-DOCUMENTAÇÃO DAS CLASSES E PACOTES DO PROJETO-

SALVADOR

2024

1. Pacote Model Classe Aluno

Este código Java define uma classe chamada Aluno, que será mapeada para uma tabela de banco de dados chamada "**Alunos**". Ele utiliza a JPA (**Java Persistence API**) com a implementação de jakarta.persistence para realizar o mapeamento objeto-relacional (ORM).

1. Anotação @Entity:

- - Indica que a classe Aluno é uma entidade JPA. Cada instância da classe corresponde a uma linha na tabela do banco de dados.

2. *Anotação @Table(name = "Alunos"):

- - Especifica o nome da tabela no banco de dados.

3. Atributos da Classe Aluno:

- - codigo: Identificador único para cada aluno, marcado com @Id para indicar que é a chave primária. A geração do valor é automatizada com a estratégia GenerationType.IDENTITY.
- - nome: Nome do aluno.
- - cpf: CPF do aluno, marcado com @Column(unique = true) para garantir que cada CPF seja único no banco de dados.
- - idade: Idade do aluno.
- - treino: Descrição do treino do aluno.
- - email: Endereço de e-mail do aluno.
- - telefone: Número de telefone do aluno.

Funcionalidades

- Persistência de Dados: Permite que instâncias da classe Aluno sejam persistidas em um banco de dados relacional.

- Gerenciamento de Identidade: Garante que cada aluno tenha um identificador único (codigo), que é gerado automaticamente.

- Encapsulamento: Protege os atributos da classe, fornecendo acesso controlado através dos métodos getters e setters.

- Mapeamento Objeto-Relacional: Facilita a integração entre a aplicação Java e o banco de dados, permitindo que operações CRUD (Create, Read, Update, Delete) sejam realizadas de maneira simplificada.

2. Pacote Controller, Classe AlunoController

Define um controlador REST para a entidade Aluno. Ele gerencia operações CRUD (Create, Read, Update, Delete) para Aluno.

- Anotações:
 - `@RestController`: Indica que a classe é um controlador REST, permitindo que ela manipule requisições HTTP.
 - `@RequestMapping("/alunos")`: Define o caminho base para todas as rotas deste controlador.
 - `@Autowired`: Injeta automaticamente a dependência do repositório Repositorio.
- Dependências:
 - Repositorio acao: Este é o repositório JPA que fornece métodos para acessar e manipular dados da entidade Aluno.

Metodos

1. `*@PostMapping("/cadastrar")`.

- **Propósito:** Cria um novo aluno.
- **Parâmetro:** Recebe um objeto Aluno no corpo da requisição HTTP (JSON).
- **Processo:** Chama o método save do repositório (acao) para salvar o aluno no banco de dados.
- **Resposta:** Retorna um ResponseEntity contendo o aluno recém-criado e um status HTTP 200 (OK).

2. `@GetMapping("/listar")`

- **Propósito:** Lista todos os alunos ordenados pelo nome.
- **Parâmetro:** Nenhum.
- **Processo:** Chama o método findAllByOrderByNome do repositório (acao) para obter a lista de alunos.
- **Resposta:** Retorna um ResponseEntity contendo a lista de alunos e um status HTTP 200 (OK).

3. `@GetMapping("/buscar/{cpf}")`

- **Propósito:** Busca um aluno pelo CPF.
- **Parâmetro:** Recebe o CPF como parte da URL (`@PathVariable`).
- **Processo:** Chama o método findByCpf do repositório (acao) para encontrar o aluno pelo CPF.
- **Resposta:**
 - Se o aluno é encontrado, retorna um ResponseEntity contendo o aluno e um status HTTP 200 (OK).
 - Se não encontrado, retorna um ResponseEntity com status HTTP 404 (N).ot Found

4. `@DeleteMapping("/excluir/{cpf}")`

- **Propósito:** Exclui um aluno pelo CPF.

- **Parâmetro:** Recebe o CPF como parte da URL (@PathVariable).
- **Processo:**
Chama o método findByCpf do repositório (acao) para encontrar o aluno pelo CPF.
- Se o aluno é encontrado, chama o método delete do repositório para removê-lo.
- **Resposta:**
- Se o aluno é encontrado e excluído, retorna um ResponseEntity com status HTTP 200 (OK).
- Se o aluno não é encontrado, retorna um ResponseEntity com status HTTP 404 (Not Found).

5. @PutMapping("/editar")

- **Propósito:** Edita as informações de um aluno existente.
- **Parâmetro:** Recebe um objeto Aluno no corpo da requisição HTTP (JSON).
- **Processo:**
- Verifica se o aluno existe no banco de dados pelo código (existsById).
- Se existir, chama o método save do repositório (acao) para salvar as novas informações.
- **Resposta**
- Se o aluno é encontrado e editado, retorna um ResponseEntity contendo o aluno ---- - editado e um status HTTP 200 (OK).
- Se o aluno não é encontrado, retorna um ResponseEntity com status HTTP 404 (Not Found).

3. Pacote serviço classe AlunoService

Anotações e Dependências:

- @Service: Indica que esta classe é um serviço do Spring, sendo gerenciada pelo Spring Container.
- @Autowired: Injeta automaticamente a dependência do repositório Repositorio.

- **Funcionabilidades**

- - Cadastro de Aluno: Permite adicionar novos alunos ao banco de dados.
- - Listagem de Alunos: Retorna uma lista de todos os alunos ordenados pelo nome.
- - Busca por CPF: Permite buscar alunos pelo CPF.
- - Exclusão de Aluno: Permite excluir um aluno pelo CPF.
- - Edição de Aluno: Permite atualizar as informações de um aluno existente.

Integração com o Controlador:

Os métodos do AlunoService são utilizados pelo controlador AlunoController para implementar as operações de cadastro, listagem, busca, exclusão e edição de alunos

4. Pacote Repository classe AlunoRepository

O repositório Repositorio define métodos específicos para a entidade Aluno, permitindo operações como busca por CPF, listagem ordenada por nome e exclusão por CPF. Estes métodos são utilizados pelo serviço AlunoService para implementar a lógica de negócio da aplicação, facilitando a manutenção e testes do código.

Os métodos do repositório Repositorio são utilizados pelo serviço AlunoService para implementar a lógica de negócio relacionada às operações CRUD. Isso ajuda a manter a separação de responsabilidades, facilitando a manutenção e testes do código.

Anotações e Extensão:

- @Repository: Indica que esta interface é um repositório do Spring, sendo gerenciada pelo Spring Container.
- JpaRepository<Aluno, Integer>: Estende a interface JpaRepository, fornecendo métodos genéricos para operações CRUD. O tipo da entidade é Aluno e o tipo da chave primária é Integer.