



*Serviço Nacional de Aprendizagem Industrial*

**PELO FUTURO DO TRABALHO**

# Repositórios

## Desenvolvimento de Sistemas

Prof. Me. Reneilson Santos

Abril/2024

# Agenda

- Repositórios
- Bibliografia



# Repositórios



# Repositórios

O objetivo da abstração do repositório Spring Data é **reduzir** significativamente a **quantidade de código** clichê **necessário para implementar camadas de acesso** a dados para vários armazenamentos de persistência.

A interface central na abstração de repositório do Spring Data é a interface ***Repository***.

Leva a classe de domínio para gerenciar, bem como o tipo de id da classe de domínio como argumentos de tipo.

Essa interface atua principalmente como uma interface de marcador para capturar os tipos com os quais trabalhar e ajudá-lo a descobrir interfaces que estendem esta.

# Outros Repositórios

- ***CrudRepository*** fornece funcionalidade CRUD sofisticada para a classe de entidade que está sendo gerenciada.
  - Permite salvar (POST), recuperar informações (GET), deletar (DELETE), além de métodos de contagem, existência, etc.
  - Permite também a criação de outros métodos utilizando filtros criados pelo usuário.
- ***PagingAndSortingRepository*** estende o *CrudRepository* adicionando paginação e ordenação.
  - Exemplo de uso:  
`Page<User> users = repository.findAll(new PageRequest(1, 20));`
- ***JpaRepository*** estende o *PagingAndSortingRepository* adicionando algumas funcionalidades do JPA como o save and flush.

# Criando Repositório

Portanto, para criarmos nosso repositório basta criarmos uma interface estendendo uma das interfaces de repositório (***Repository***, ***CrudRepository***, ***JpaRepository***, ***PagingAndSortingRepository***) e dentro podemos adicionar novos métodos para criar queries específicas.

Podemos criar um pacote para repositórios dentro de nosso projeto ou um pacote dentro dos controles.

No SpringBoot adicionamos à nossa interface de repositório a annotation **@Repository** para fazê-la ser encontrada pela aplicação e poder ser utilizada como repositório.

```
public interface PetRepository extends JpaRepository<Pet, Long> {}
```

```
public interface UserRepository extends JpaRepository<User, Long> {}
```

```
public interface TagRepository extends JpaRepository<Tag, Long> {}
```

```
public interface CategoryRepository extends JpaRepository<Category, Long> {}
```

```
public interface AdministratorRepository extends JpaRepository<Administrator, Long> {}
```



# Bibliografia



# Bibliografia

- JPA Repositories  
(<https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html>)
- Accessing Data with JPA (<https://spring.io/guides/gs/accessing-data-jpa/>)
- Spring Data Partial Update  
(<https://www.baeldung.com/spring-data-partial-update>)
-