

Para saber mais: Queries entre relacionamentos

Já vimos duas formas como executar consultas com Spring Data no repositório:

- *Derived Query Methods*
- métodos anotados com `@Query` .

Aprendemos que ao usar o *Derived Query Methods* o JPQL é gerado dinamicamente (*ou derivado*) baseado no nome do método. Não mostramos no vídeo mas claro que isso também funciona para consultas que acessam os relacionamentos!

Por exemplo, veja o método abaixo onde estamos procurando funcionários pela descrição do cargo:

```
//deve estar no repositório do funcionário  
List<Funcionario> findByCargoDescricao(String descricao);
```

[COPIAR CÓDIGO](#)

Repare que usamos `findBy` para depois definir o caminho no relacionamento `CargoDescricao` (a descrição é um atributo dentro do `Cargo`). O método é análogo ao JPQL abaixo:

```
@Query("SELECT f FROM Funcionario f JOIN f.cargo c WHERE  
c.descricao = :descricao")  
List<Funcionario> findByCargoPelaDescricao(String  
descricao);
```

[COPIAR CÓDIGO](#)

Ficou claro?

Agora imagina se precisa pesquisar pela descrição mas da `UnidadeTrabalho`. A primeira ideia seria usar o nome `findByUnidadeTrabalhosDescricao(String descricao)` como discutimos.

No entanto temos o problema que o nome da entidade `UnidadeTrabalho` é composto de duas palavras. Para separar claramente o nome da entidade do atributo devemos usar o caracter `_`. Veja a assinatura do método então:

```
List<Funcionario> findByUnidadeTrabalhos_Descricao(String descricao);
```

[COPIAR CÓDIGO](#)

Também analisa a mesma pesquisa com JPQL e `@Query` :

```
@Query("SELECT f FROM Funcionario f JOIN f.unidadeTrabalhos u WHERE u.descricao = :descricao")  
List<Funcionario> findByUnidadeTrabalhos_Descricao(String descricao);
```

[COPIAR CÓDIGO](#)

Repare que nesse exemplo, bastante simples ainda, o nome do método já cresceu e usa uma nomenclatura fora do padrão Java. Isso é uma desvantagem dos *Derived Query Methods*.

Caso que precise consultas um pouco mais complexas, por exemplo usando relacionamentos e vários parâmetros, dê a preferência aos métodos com `@Query` para não prejudicar o entendimento pois os nomes dos métodos vão ficar muito longos para definir todos os critérios de busca. Tudo bem?

