



Um dos recursos mais interessantes do Spring Data são as **“derived queries”**, basicamente são métodos abstratos que **seguem uma sintaxe** que permite executar **queries padrões através** das implementações **FORNECIDAS PELO SDJ**.

Por exemplo, se eu quiser encontrar uma **lista de funcionários pelo nome**, eu preciso simplesmente fazer um método usando a **sintaxe** de **findByXXX** :

```
@Repository
public interface FuncionarioRepository extends CrudRepository<Funcionario, Long> {

    List<Funcionario> findByNome(String nome);
}
```

Entre o **find** e o **By** podemos usar **qualquer prefixo**, o que te permite **especificar um pouco mais as queries**, por exemplo eu poderia fazer **“findFuncionariosByNome”**. As **sintaxes** dos métodos com os **tipos mais comuns de query** são inúmeras, portanto, vale a pena conferir a **doc** sempre que quiser usar o recurso:

<https://docs.spring.io/spring-data/jpa/docs/1.5.0.RELEASE/reference/html/jpa.repositories.html>

| Keyword | Sample | JPQL snippet |
|--------------------|---|--|
| And | findByLastNameAndFirstName | ... where x.lastname = ?1 and x.firstname = ?2 |
| Or | findByLastNameOrFirstName | ... where x.lastname = ?1 or x.firstname = ?2 |
| Is, Equals | findByFirstName, findByFirstNameIs, findByFirstNameEquals | ... where x.firstname = ?1 |
| Between | findByStartDateBetween | ... where x.startDate between ?1 and ?2 |
| LessThan | findByAgeLessThan | ... where x.age < ?1 |
| LessThanEqual | findByAgeLessThanEqual | ... where x.age <= ?1 |
| GreaterThan | findByAgeGreaterThan | ... where x.age > ?1 |
| GreaterThanEqual | findByAgeGreaterThanEqual | ... where x.age >= ?1 |
| After | findByStartDateAfter | ... where x.startDate > ?1 |
| Before | findByStartDateBefore | ... where x.startDate < ?1 |
| IsNull | findByAgeIsNull | ... where x.age is null |
| IsNotNull, NotNull | findByAge(Is)NotNull | ... where x.age not null |

Uma coisa que os **repositórios SDJ** permitem é usar **JPQL** em **“query methods/derived queries”**, basta usar a anotação **@Query(“xxxx”)** passando a **JPQL**, nesse caso a **sintaxe do método não importa**:

```
@Repository
public interface FuncionarioRepository extends CrudRepository<Funcionario, Long> {

    @Query("from Funcionario f JOIN FETCH f.cargo WHERE f.nome = :nome")
    List<Funcionario> findByNome(String nome);

    @Query("from Funcionario f JOIN FETCH f.cargo WHERE f.salario >= :salario")
    List<Funcionario> buscarPorSalarioMaiorQue(BigDecimal salario);
}
```

Também é possível fazer **queries nativas**:

```
@Query(value = "SELECT * FROM funcionarios WHERE funcionarios.data_contratacao >= :dataContratacao",
nativeQuery = true)
List<Funcionario> buscarPorDataContratacao(LocalDate dataContratacao);
```