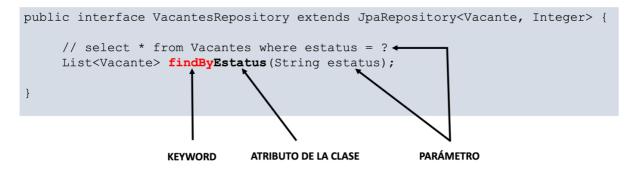
MÓDULO XI – SPRING DATA JPA III - QUERY METHODS

¿Qué son los query methods?

Son métodos que permiten encontrar información (SELECT) en la base de datos y son **DECLARADOS** en la interfaz del repositorio.

Nota: Estos métodos solo se **DECLARAN** en la interfaz, Spring Data JPA realiza la implementación del método, dependiendo del **NOMBRE DEL METODO y los PARAMETROS que recibe.**

Ejemplo: Método para recuperar las entidades de tipo Vacante, filtradas por el campo estatus:



Para que JPA detecte y cree las sentencias SQL automáticamente debemos seguir una regla sintáctica:

El nombre del método debe ser una mezcla de palabras clave, en combinación con el nombre del argumento del método. Existen muchas palabras reservadas para poder crear métodos de consulta mas complejos. Con el fin de ampliar la información al alumno, sobre los Query Methods disponibles, se adjunta la siguiente tabla:

Table 4. Supported keywords inside method names		
Keyword	Sample	JPQL snippet
		Java Persistence Query Language
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
Like	findByFirstnameLike	where x.firstname like ?1
NotLike	findByFirstnameNotLike	where x.firstname not like ?1
StartingWith	findByFirstnameStartingWith	where x.firstname like ?1 (parameter bound with
		appended %) → where x.firstname like '?%'
EndingWith	findByFirstnameEndingWith	where x.firstname like ?1 (parameter bound with
		prepended %) → where x.firstname like `%?'
Containing	findByFirstnameContaining	where x.firstname like ?1 (parameter bound
		wrapped in %) → where x.firstname like '%?%'
OrderBy	findByAgeOrderByLastnameDesc	where x.age = ?1 order by x.lastname desc
Not	findByLastnameNot	where x.lastname <> ?1
In	<pre>findByAgeIn(Collection<age> ages)</age></pre>	where x.age in ?1
NotIn	<pre>findByAgeNotIn(Collection<age> age)</age></pre>	where x.age not in ?1
True	findByActiveTrue()	where x.active = true
False	findByActiveFalse()	where x.active = false
IgnoreCase	findByFirstnameIgnoreCase	where UPPER(x.firstame) = UPPER(?1)
First	findFirstBy	limit 1
Top	findTopBy	limit 1
First	findFirst15By	limit 15
Top	findTop15By	limit 15
Top	findTop20ByOrderByAgeDesc	order by x.age desc limit 20

Resumiendo:

El nombre del método debe seguir una serie de reglas para que de él se pueda derivar la consulta:

La consulta debe comenzar por:

- find...By
- read...By
- query...By
- count...By
- get...By

By es el delimitador para definir los criterios.

Ejemplo

- Definimos métodos en nuestro repositorio.
- Dichos métodos no necesitan implementación (se la da Spring Data).

Query SQL

SELECT * FROM productos WHERE id categoria = ? ORDER BY nombre ASC;

Query Methods

findByIdCategoriaOrderByNombreAsc(int idCategoria);