

CONSULTAS

MÚLTIPLES ESTRATEGIAS

- Nombre del método
- JPQL
- Criteria API
- QueryByExample
- Specifications

NOMBRE DEL MÉTODO

- Métodos en repositorios
- Debemos seguir una convención
- Spring genera las consultas para estos métodos

NOMBRE DEL MÉTODO

- Algunas pistas
 - Comenzar por *findBy*
 - Propiedades de la entidad
 - And | Or

MÉTODOS DE CONSULTA

- Sujeto y predicado

`List<Person> findByLastnameIgnoreCase(...);`

Diagram illustrating the components of the method call:

- `List<Person>`: tipo de retorno
- `findByLastnameIgnoreCase`: sujeto
- `(...)`: predicado

NOMBRE DEL MÉTODO

- Por ejemplo
 - *findByPrecio(double precio)*
 - *findByNombre(String nombre)*
 - *findByNombreIgnoreCase(...)*

TIPOS DE RETORNO

- T (la entidad), Optional<T>
- List<T>, Collection<T>, Iterator<T>
- Slice<T>, Page<T>
- void, primitivos, wrappers
- ...

KEYWORDS PARA PREDICADOS

- And, Or
- After, Before
- Containing, EndingWith, StartingWith
- GreaterThan, GreaterThanEqual,
LessThan, LessThanEqual,

NOMBRE DEL MÉTODO

- Múltiples posibilidades
- <https://docs.spring.io/spring-data/jpa/reference/repositories/query-keywords-reference.html>

JPQL

- Como un SQL de entidades
- Opera sobre clases y atributos
- No opera sobre tablas o columnas

JPQL

- Con JPA a secas es más complicado de ejecutar
- Con Spring Data JPA

```
@Query("...")  
List<Entity> method();
```

SINTAXIS JPQL

- SELECT e FROM Entity e WHERE ...

SELECT p FROM

Producto p

WHERE p.nombre LIKE %?1