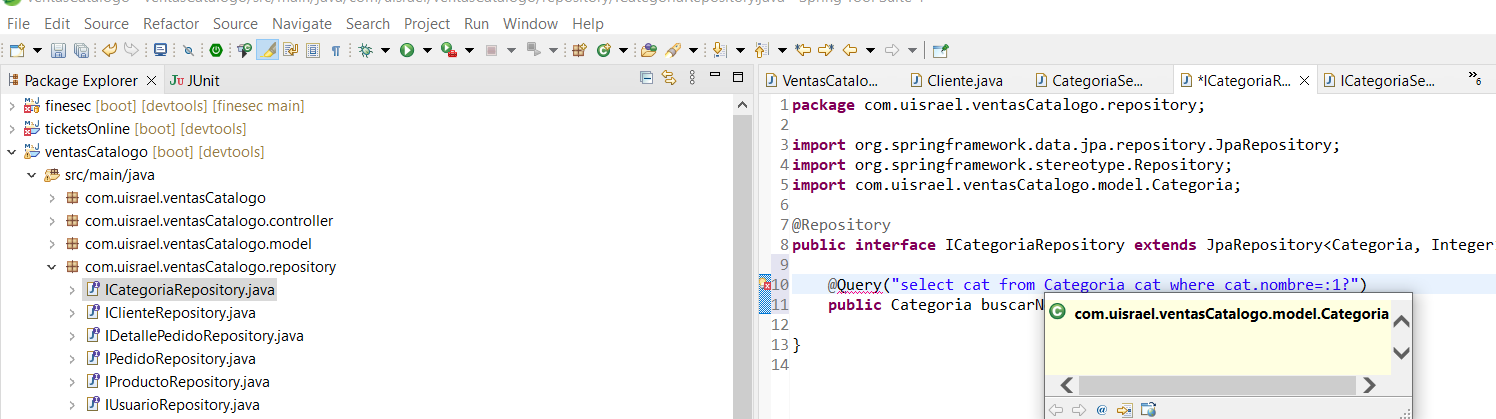
Requisitos:

Haber creado la arquitectura del proyecto  
Haber creado la arquitectura MVC  
Haber creado la conexión a la BDD   
Haber hecho un crud

**PERSISTENCIA: JPA**

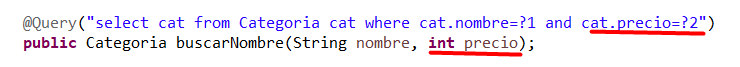
**Métodos aparte** como por ejemplo buscarPorNombre se tiene que hacer de la siguiente forma:

Vamos a buscar categoría por nombre. Vamos al repository

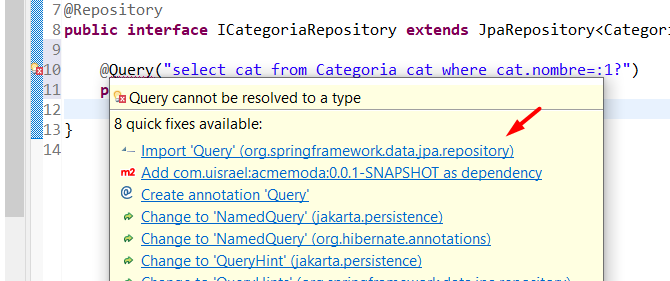




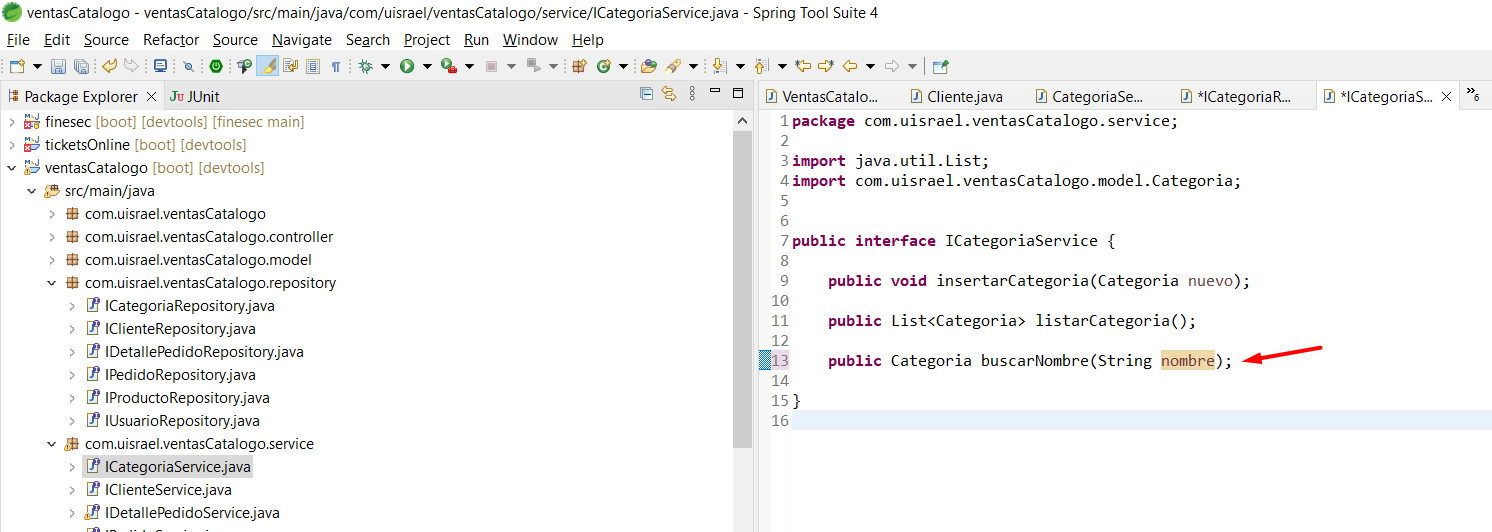
Esto es una consulta por posición. **Select cat** es lo mismo que decir **select \*** from Categoria, (Haciendo referencia al Modelo en Java equivale a hacer referencia a la talba en SQL). Where **cat.nombre** (Java) es decir where **nombre (SQL)**. Where cat.nombre=?1 Quiere decir la posición de los parámetros de la consulta. Digamos queremos listar por nombre y precio:



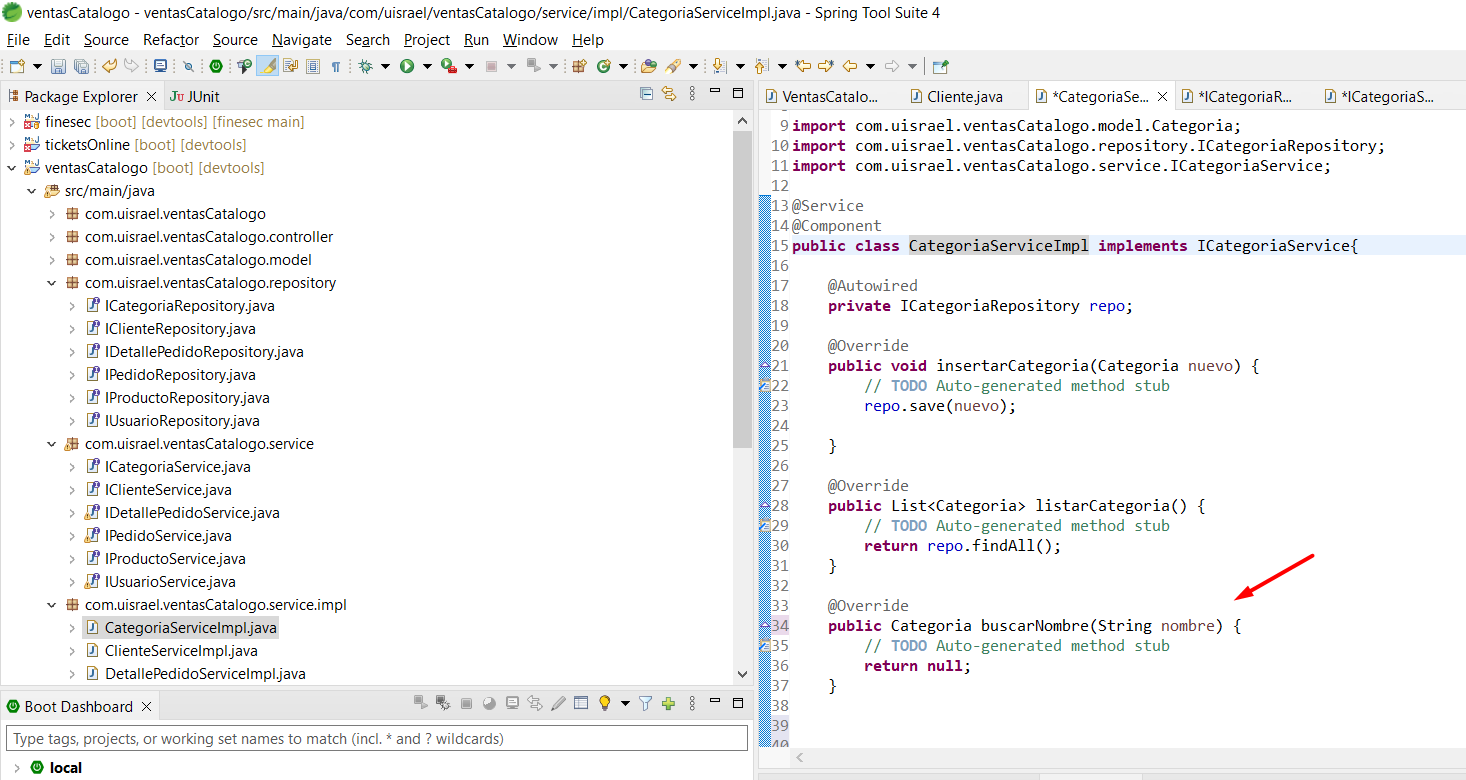
Por último importamos:



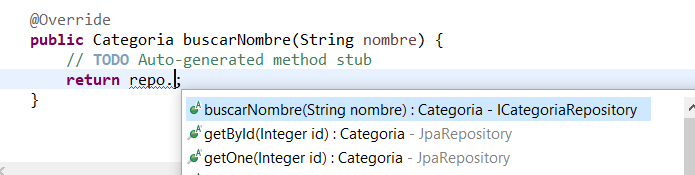
Ahora vamos al service y validamos que esté nuestro método ahí creado



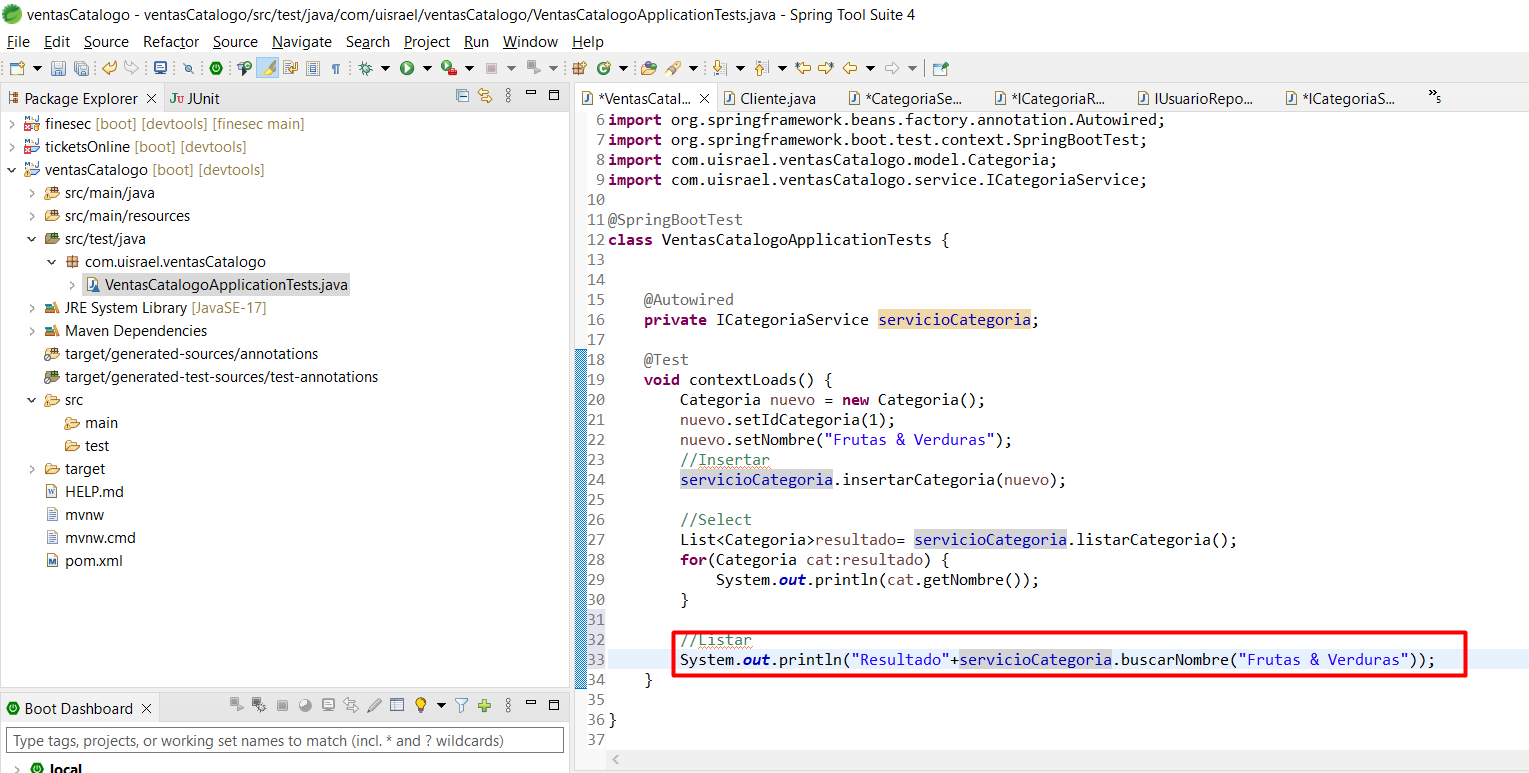
Y por último al implements para también validar que esté allí



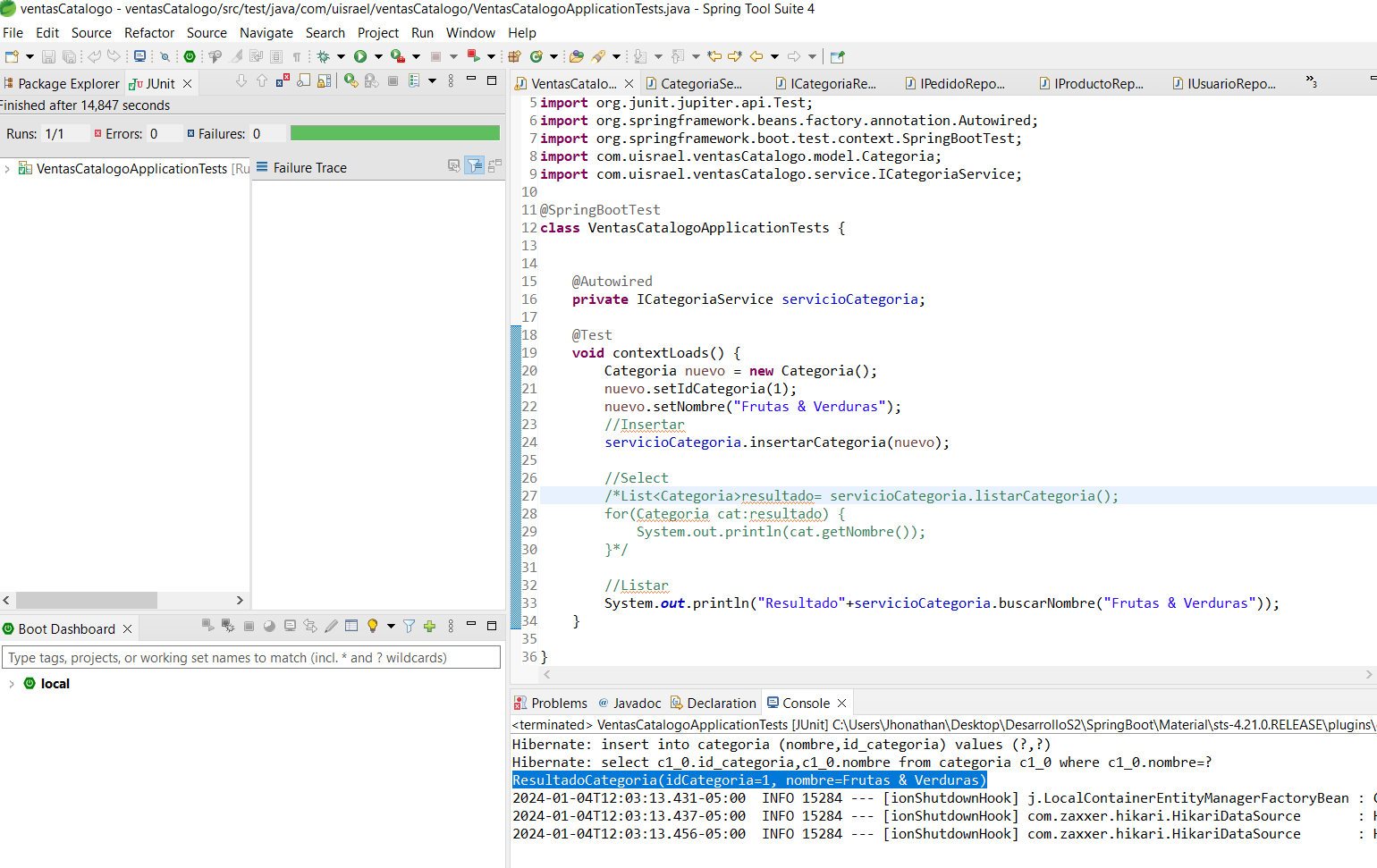
Aquí modificamos:



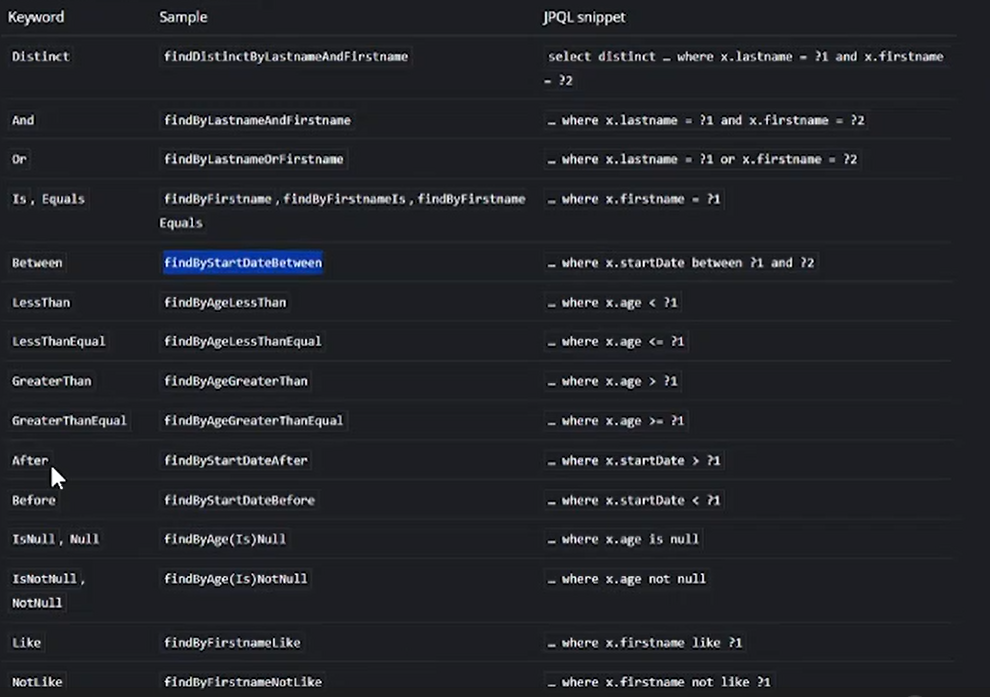
Por último vamos al test:



Resultado:



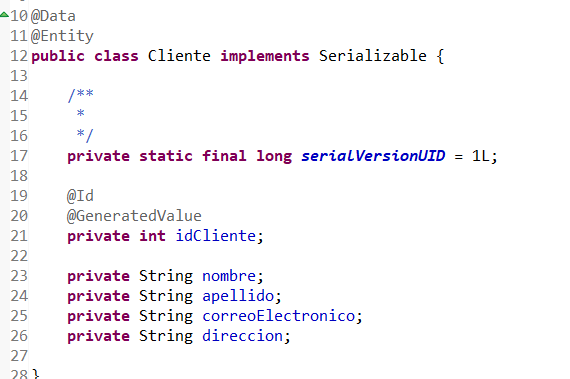
Otras consultas que podríamos utilizar puede ser:



**RELACIONES**

Se tiene que hacen dos direcciones, origen y destino.

Hasta ahora nuestros modelos tienen datos propios:

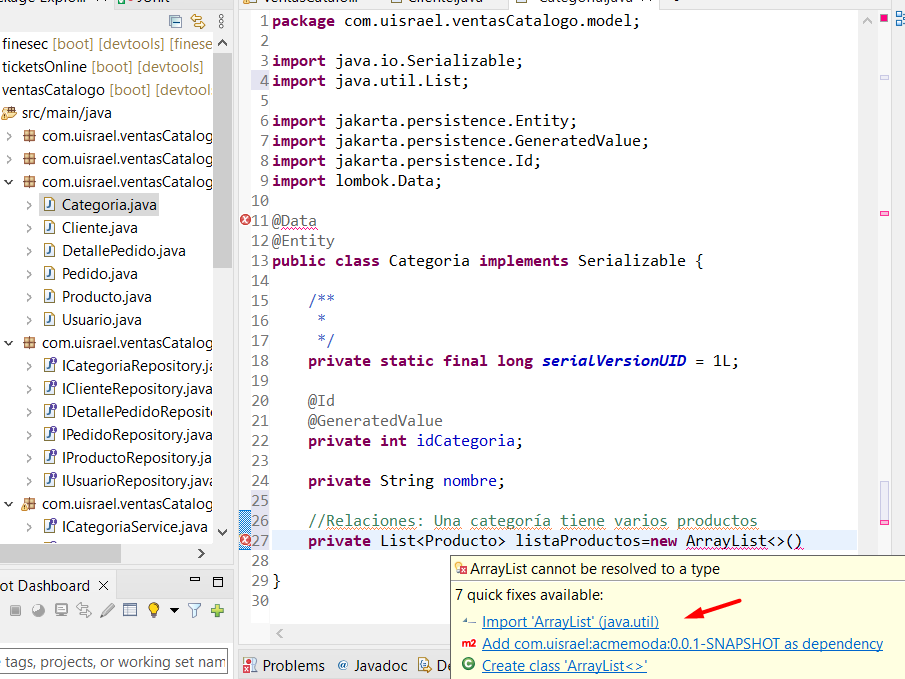


Vamos a relacionar producto con categoría

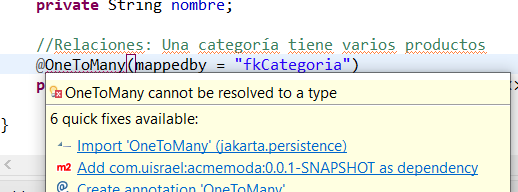
¿Cuál es la tabla origen? Cualquiera  
¿Una categoría puede tener varios productos? Si  
¿Un producto puede tener varias categorías? No

La relación es de Categoría a productos **1 a n**

Por lo tanto en el modelo Categoría:

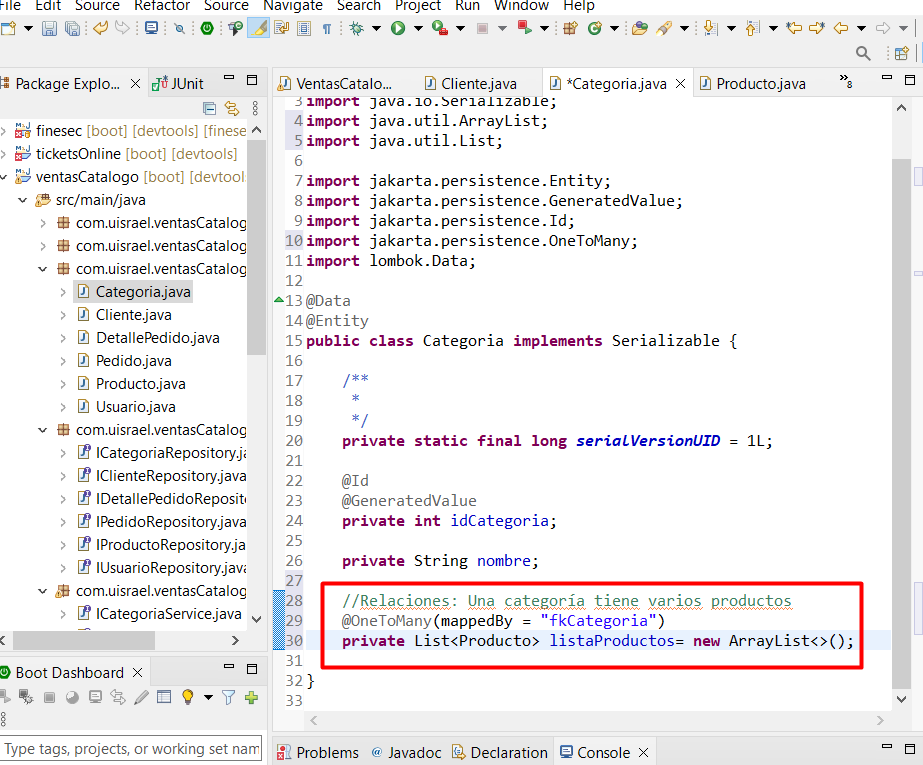


Luego usamos el @OneToMany



Le mandamos el id de nuestro modelo Categoría porque esa va a hacer la conexión con la otra clase.

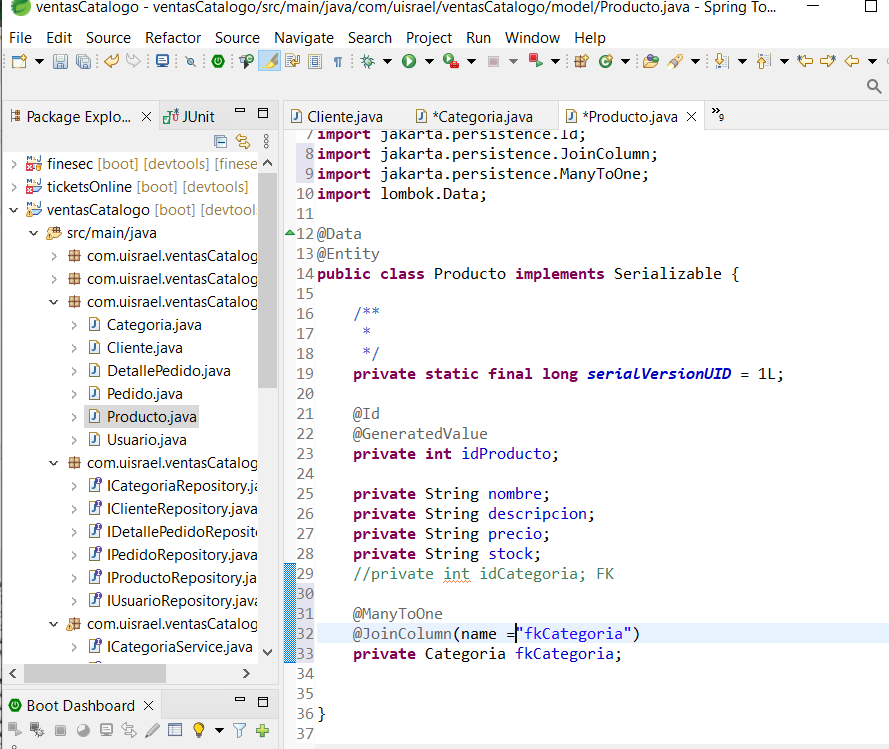
Resultado:



TRUCO: Las relaciones OneToMany siempre tienen un List

Vamos con la relación de venida:

Vamos a Producto



Y establecemos la FK, y adicionalmente el @JoinColumn para llamar así a la columna a nivel de base de datos.

Con eso hemos creado la relación

TRUCO: Las relaciones ManyToOne siempre tienen un JoinColiumn y un objeto (en este caso Categoria)