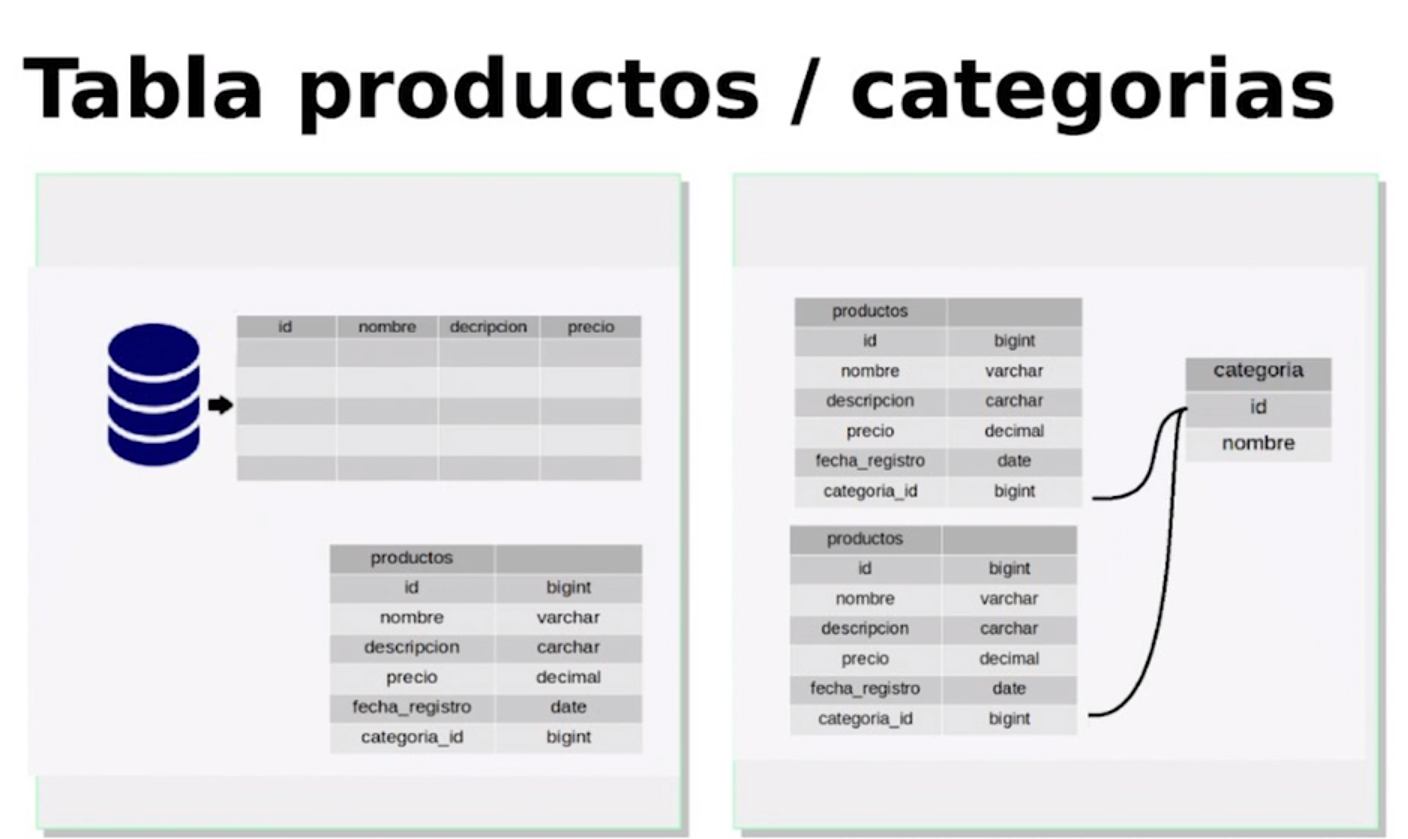
Curso de  
**Java y JPA: consultas avanzadas, rendimiento y modelos complejos**

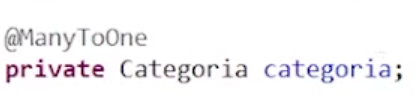
**Mas Relacionamientos:**

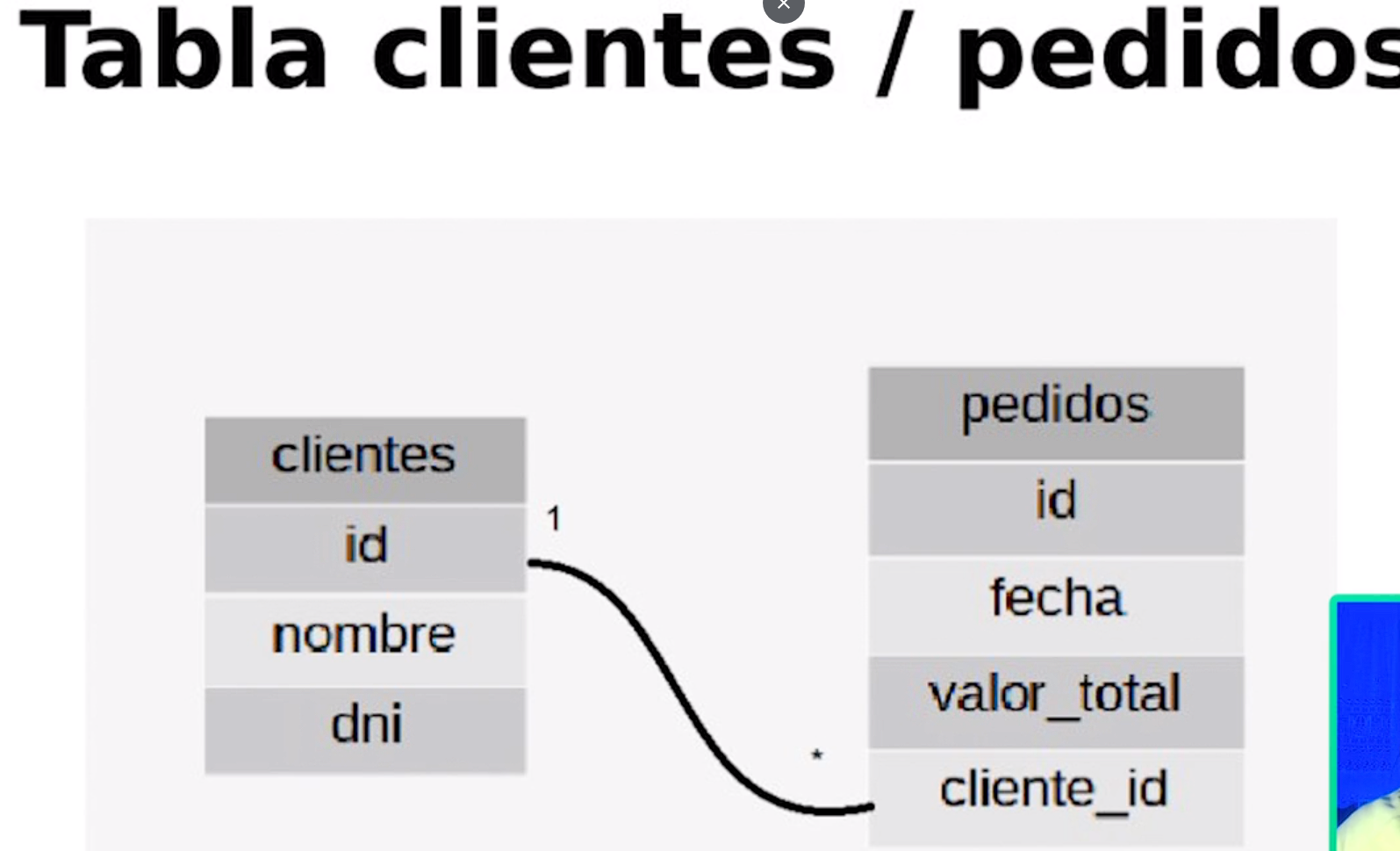
Mapeando nuevas entidades:





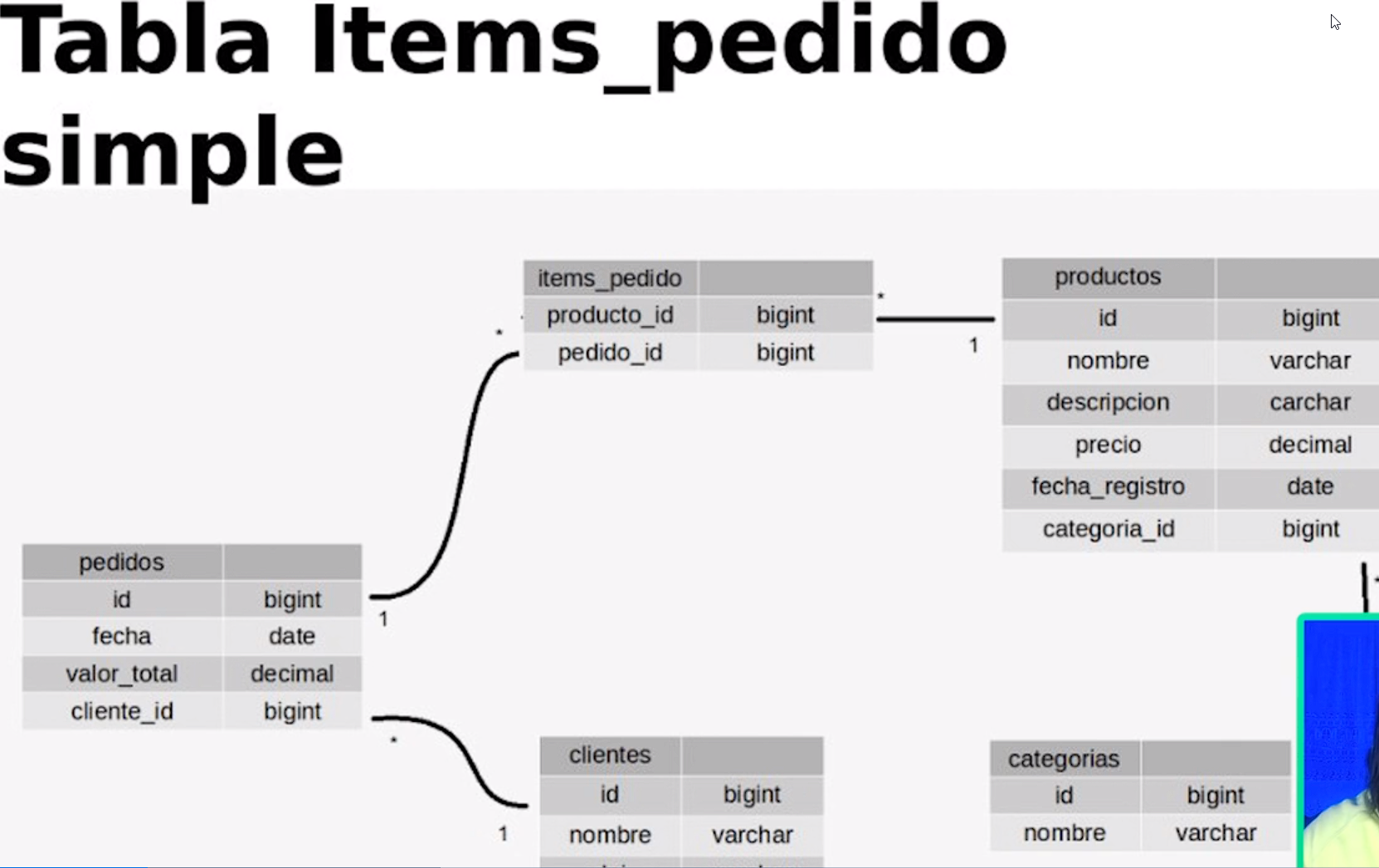
LAS ENTIDADES SE REPRESENTAN COMO CLASES MODELO

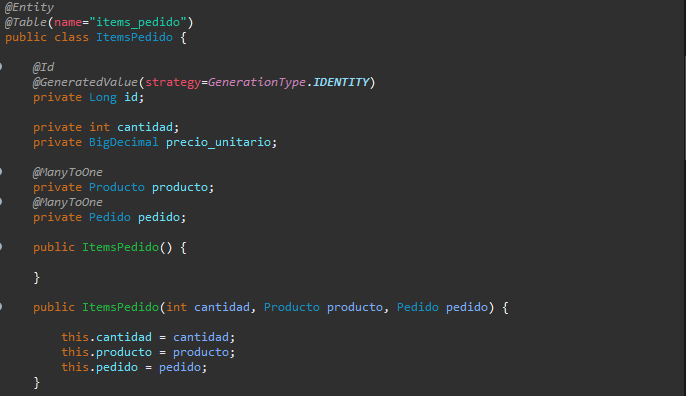






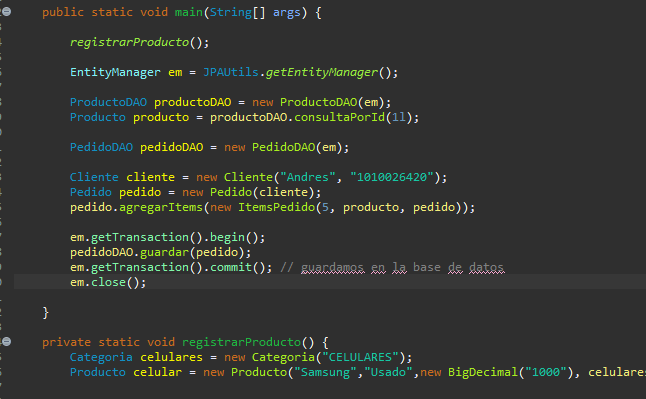
Relacionamientos Many-To-Many:

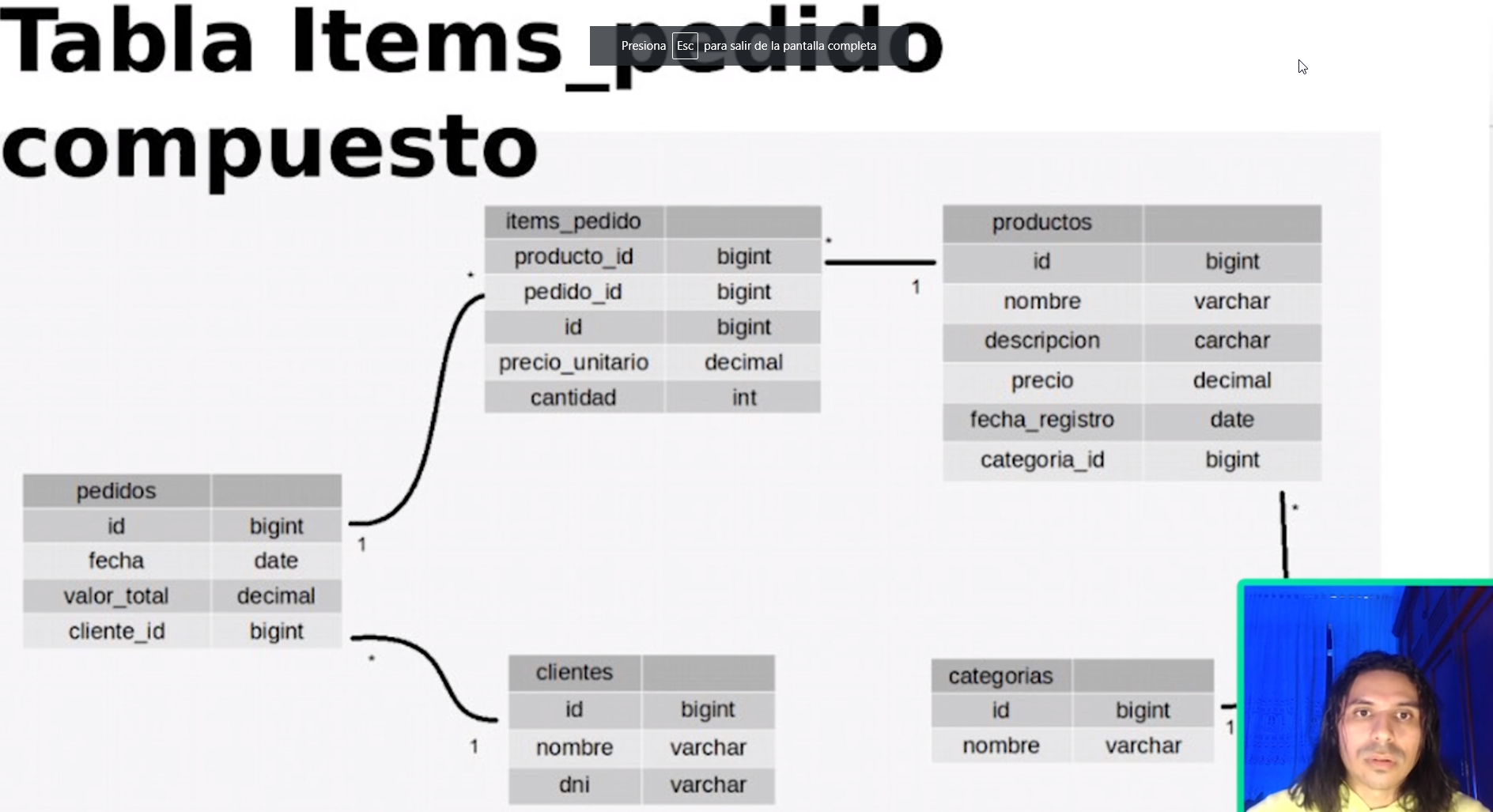




Relacionamientos Bidireccionales:

Test de relacionamiento Bidireccional:





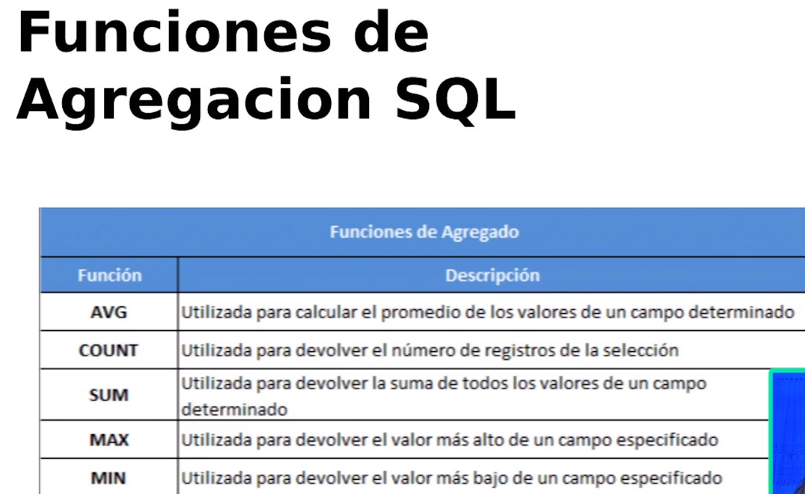
Lo que Aprendimos:

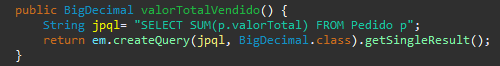
En esta aula, aprendiste:

* Mapeo de nuevas entidades en la aplicación según el modelado de la base de datos.
* Mapeo de una relación con cardinalidad de muchos a muchos.
* Mapeo de una relación bidireccional.
* Cómo persistir entidades que tienen relaciones bidireccionales.

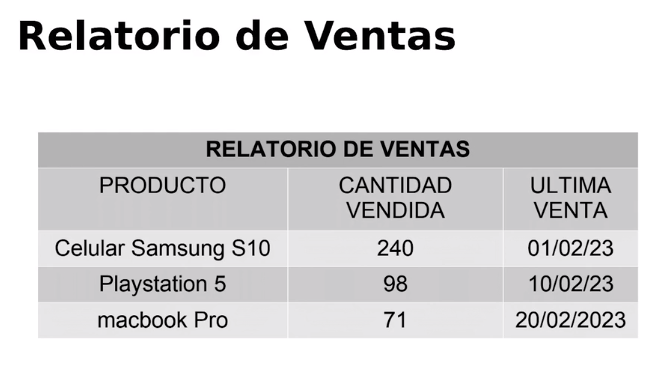
**Consultas Avanzadas:**

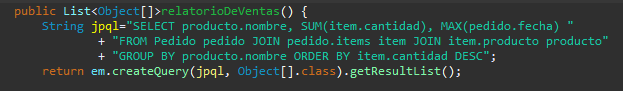
Consultas con funciones de agregación:



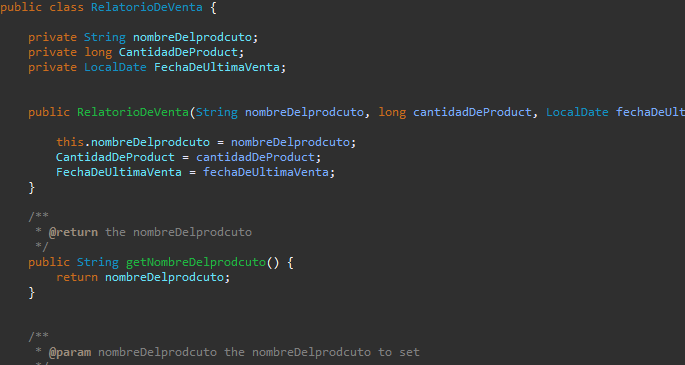


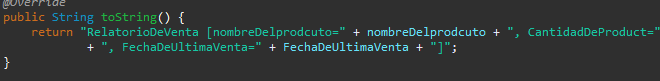
Consultas para Relatorio:

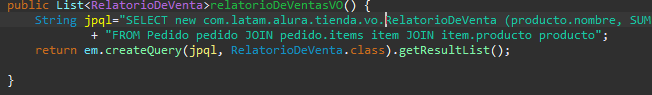


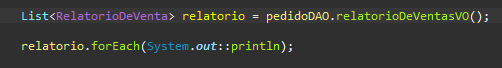


Consultas con Select New:





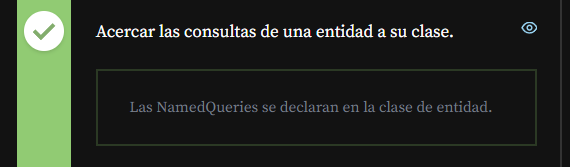




Utilizando Named Queries:







Lo que Aprendimos:

En esta aula, aprendiste:

* Realización de consultas utilizando funciones de agregación como min, max, avg y sum;
* Cómo escribir consultas de informes usando JPQL;
* Cómo usar el recurso seleccionado nuevo en consultas JPQL;
* Realización de consultas mediante consultas con nombre.

**Performance de consultas:**

Entendiendo Lazy/Eager:

EAGER:

Todo lo que tenga ToONE hará un JOIN de todos esos atributos o entidades.



LAZY:

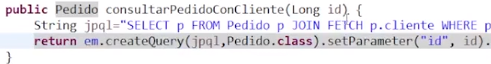
Todo lo que tenga ToMany hará un JOIN con esos atributos o entidades que sean pedidos.

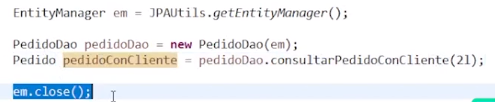


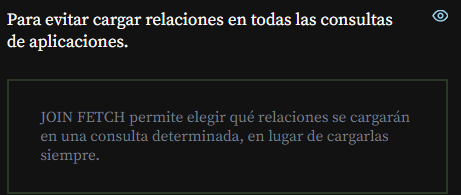


Traer los recursos únicamente cuando sean necesarios.

Consultas con Join Fetch:







Lo que Aprendimos:

En esta aula, aprendiste:

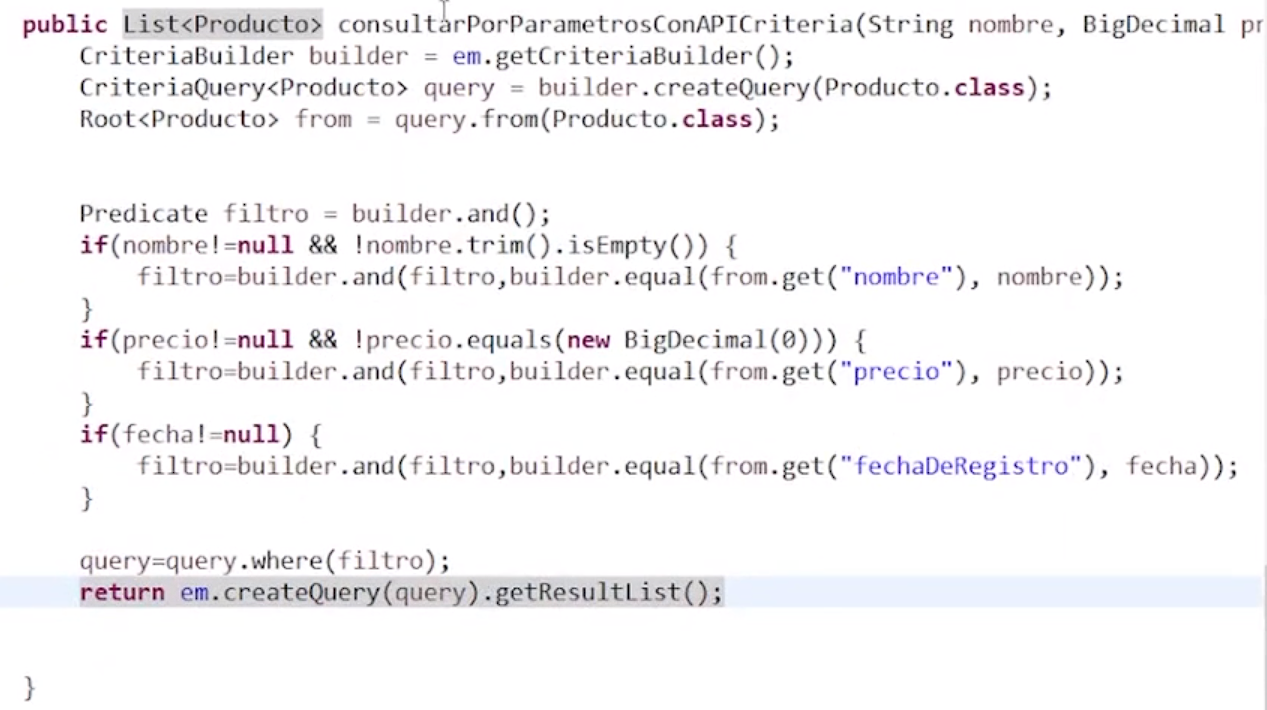
* Cómo funcionan las estrategias EAGER y LAZY, en consultas de entidades que tienen relaciones;
* Por qué JPA podría lanzar LazyInitializationException en ciertas situaciones;
* Buenas prácticas en la carga de entidades con relaciones;
* Cómo realizar consultas programadas con la función de búsqueda de combinación.

**Criteia API:**

Consultas con parametros dinámicos:



Consultas con Criteria API:



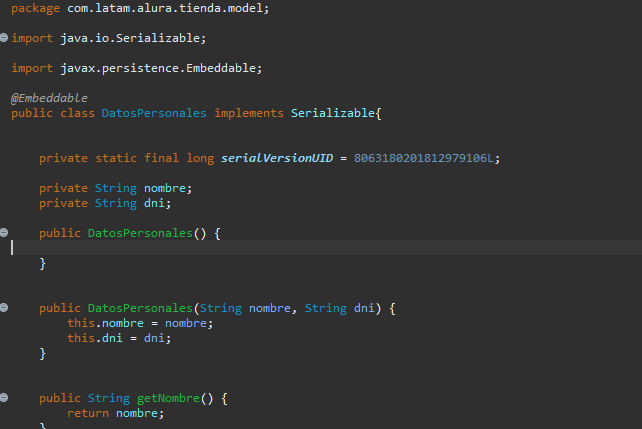
Lo que Aprendimos:

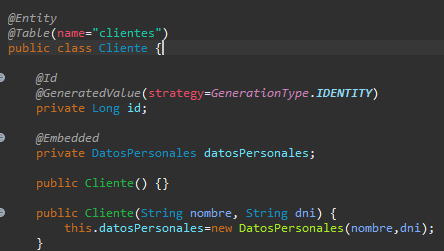
En esta aula, aprendiste:

* Cómo realizar consultas JPQL con parámetros opcionales;
* Cómo funciona la API de criterios JPA;
* Cómo realizar una consulta con parámetros opcionales a través de Criteria API.

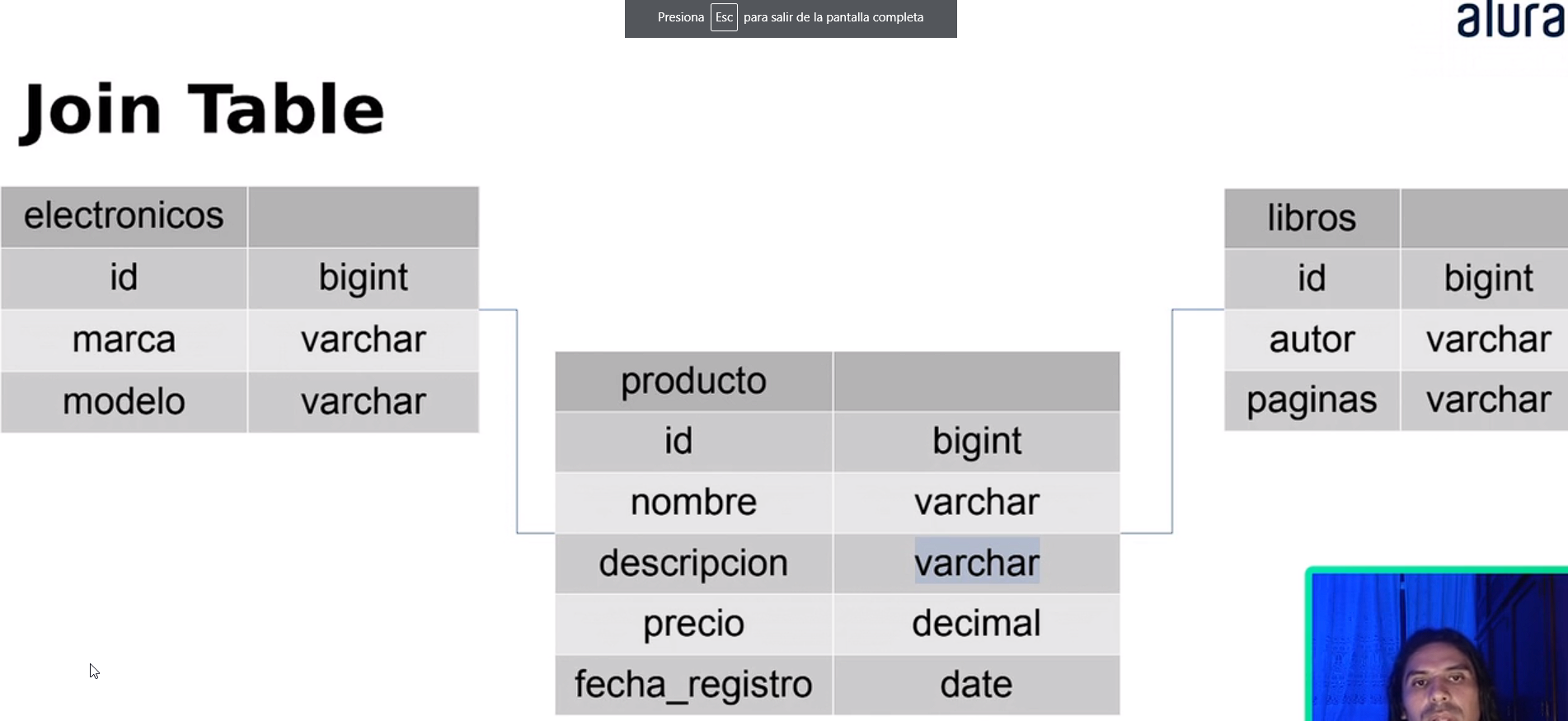
**Otros tópicos:**

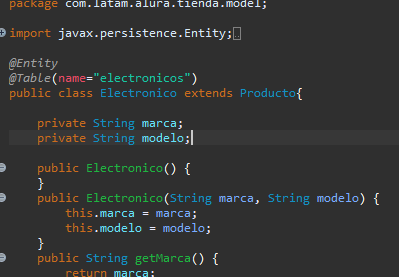
Simplificando entidades con Embeddable:

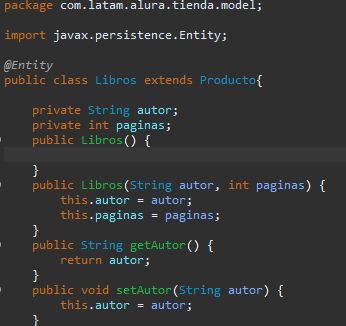




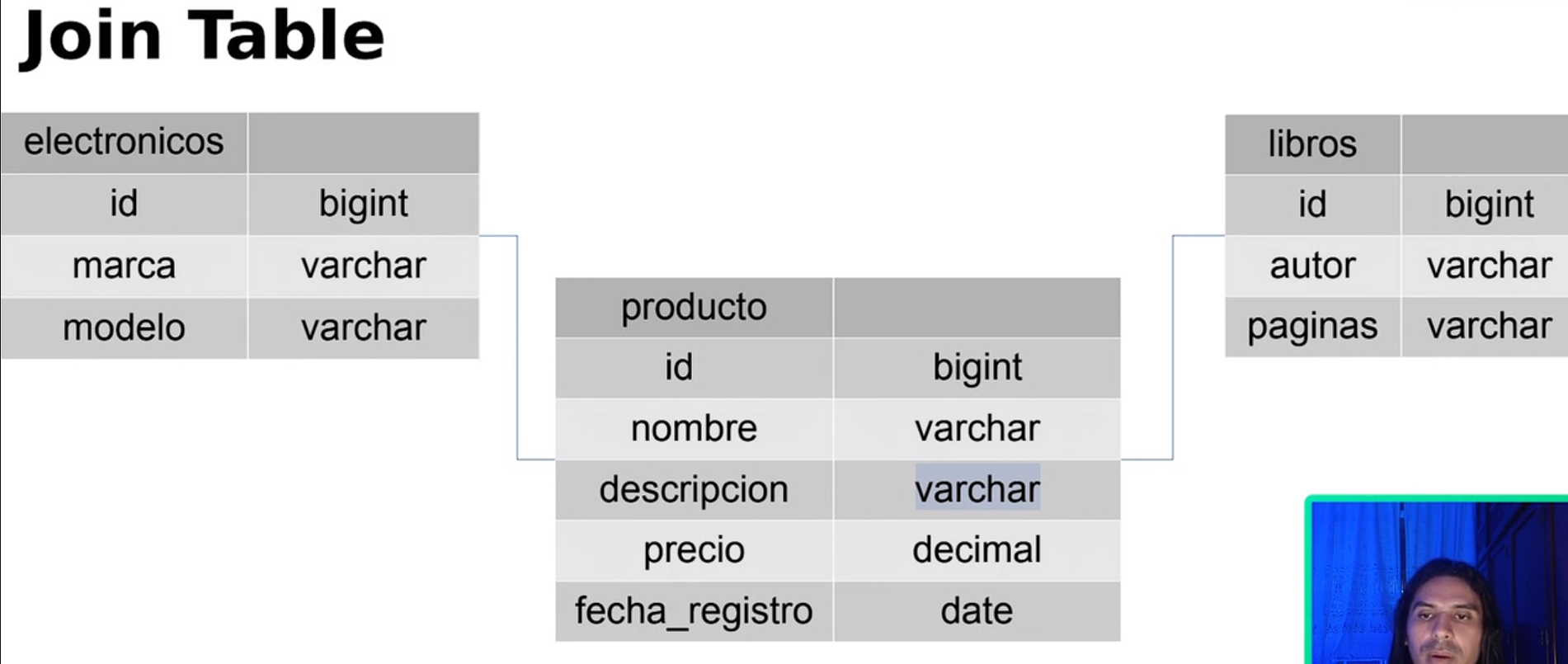
Mapeamiento de Herencias:

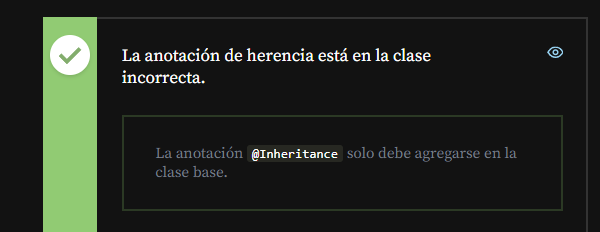




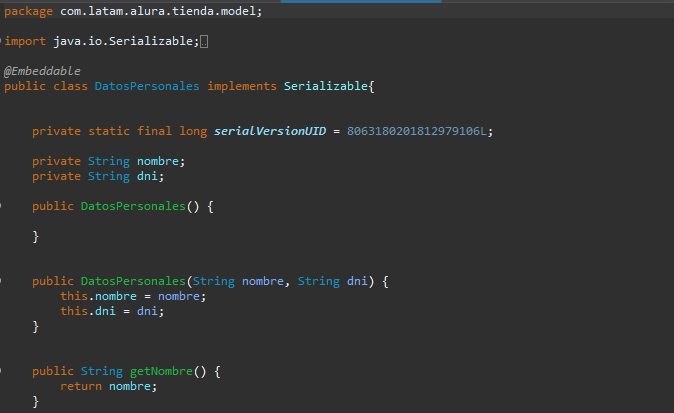


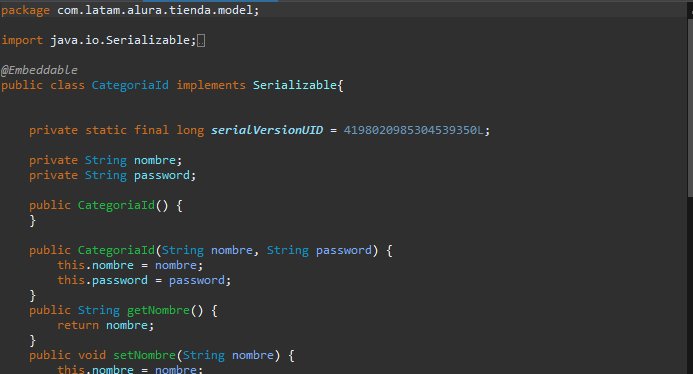


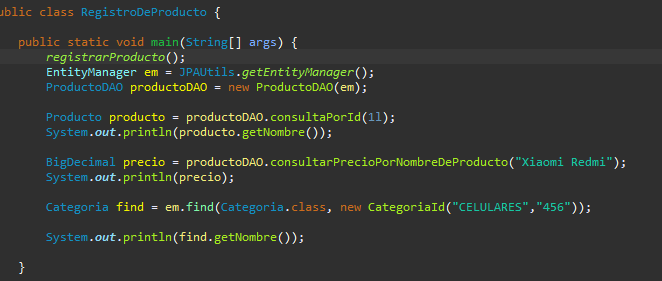




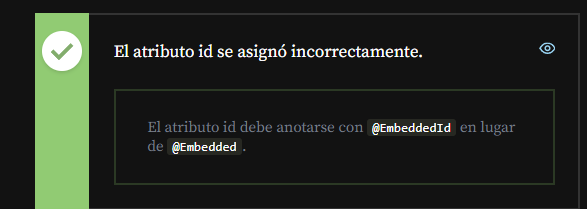
Mapeamiento de llaves compuestas:











Lo que Aprendimos:

En esta aula, aprendiste:

* Usar anotaciones @Embeddable y @Embedded para organizar el código de una entidad;
* Realizar herencia entre entidades con las estrategias SINGLE\_TABLE y JOINED;
* Cómo asignar una clave compuesta con la anotación @EmbeddedId.