

# ENTIDADES

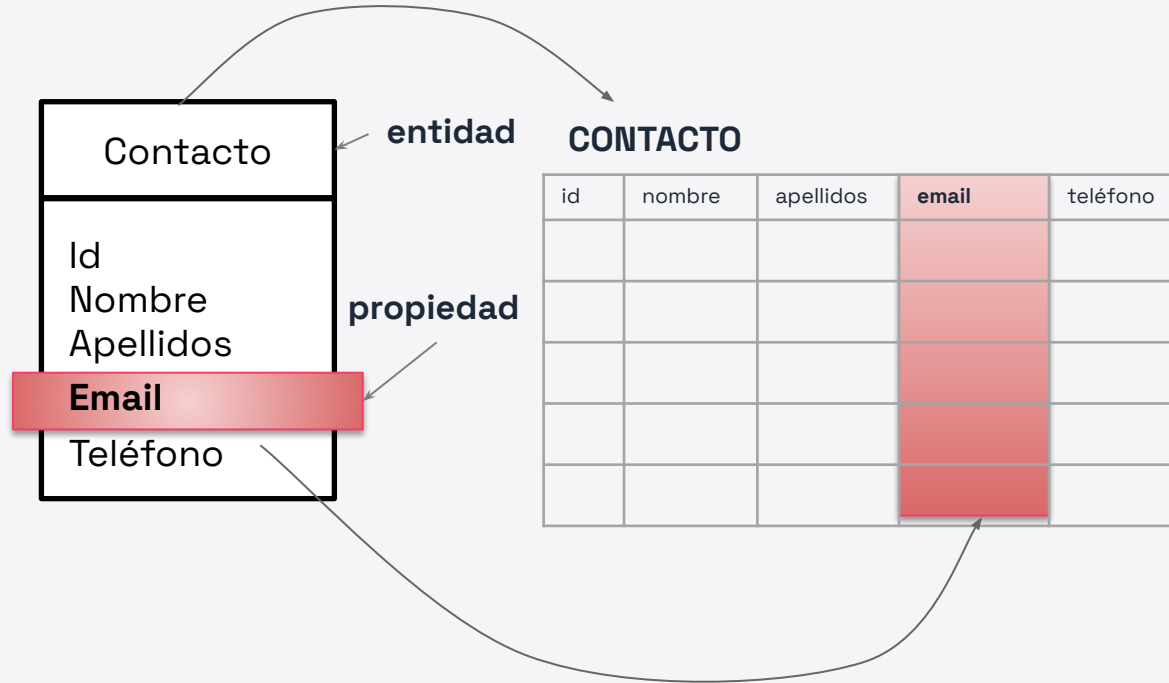
# ENTIDADES

- Son los objetos *almacenados* en la base de datos
- Objeto ↔ Tabla
- Clases POJO (@Entity)
- Identificador (@Id)

# IDENTIFICADOR GENERADO

- Valores autogenerados vs. valores de dominio
- *@GeneratedValue*
  - AUTO: mejor elección
  - SEQUENCE, IDENTITY, TABLE, UUID

# MAPEO DE ENTIDAD A TABLA



# ¿QUIÉN CREA EL DDL?

- Spring Data JPA
  - Bueno JPA
    - Bueno, realmente Hibernate
- El programador

# HIBERNATE GENERA EL DDL

- `spring.jpa.hibernate.ddl-auto`
  - none, create, drop, create-only, create-drop, validate, update
- `spring.jpa.properties.hibernate.hbm2ddl.auto`

# ALGUNAS PROPERTIES

- Valores de Spring Data JPA en **application.properties**
- Prefijos
  - **spring.datasource.XXXX**
  - **spring.jpa.XXX**

# ASOCIACIONES

- Las entidades no están aisladas
  - Producto - Categoría
  - Producto - LineaVenta
  - LineaVenta - Venta
  - ...



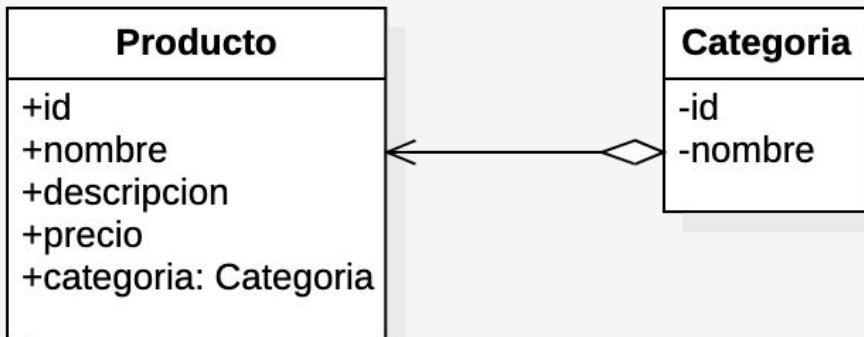
# MULTIPLICIDAD

- ManyToOne / OneToMany
- OneToOne
- ManyToMany

# DIRECCIONALIDAD

- Unidireccionales
  - Desde la entidad A hacia B
- Bidireccionales
  - Desde la entidad A hacia B, desde B hacia A

# MANYTOONE



# MANYTOONE

- Anotación @ManyToOne
- @JoinColumn
  - nombre de la columna y FK
  - tipo de dato
  - ...

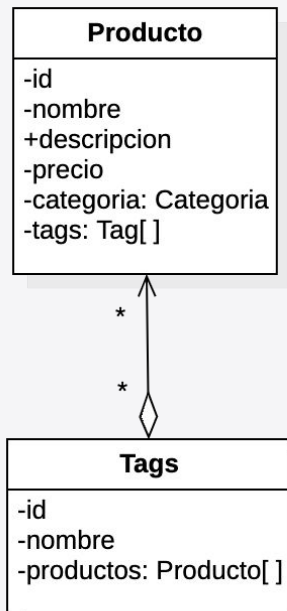
# MANYTOMANY

- Una instancia de A se relaciona con muchas de B, y viceversa
  - Proyecto - Empleado
  - Usuario - Producto (me gusta)

# MANYTOMANY

- Unidireccional
- Bidireccional
- Lo que mejor convenga a la semántica del problema

# MANYTOMANY



# ¿QUÉ TIPO DE COLECCIÓN UTILIZAR?

- **Set**
  - Genera el DDL adecuado
  - Puede que peor rendimiento
  - No contiene duplicados