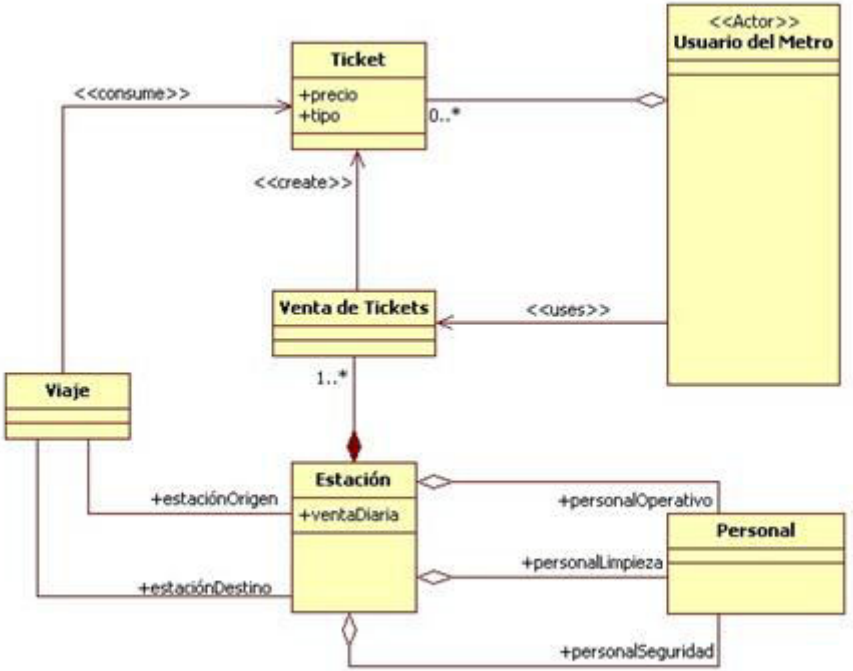
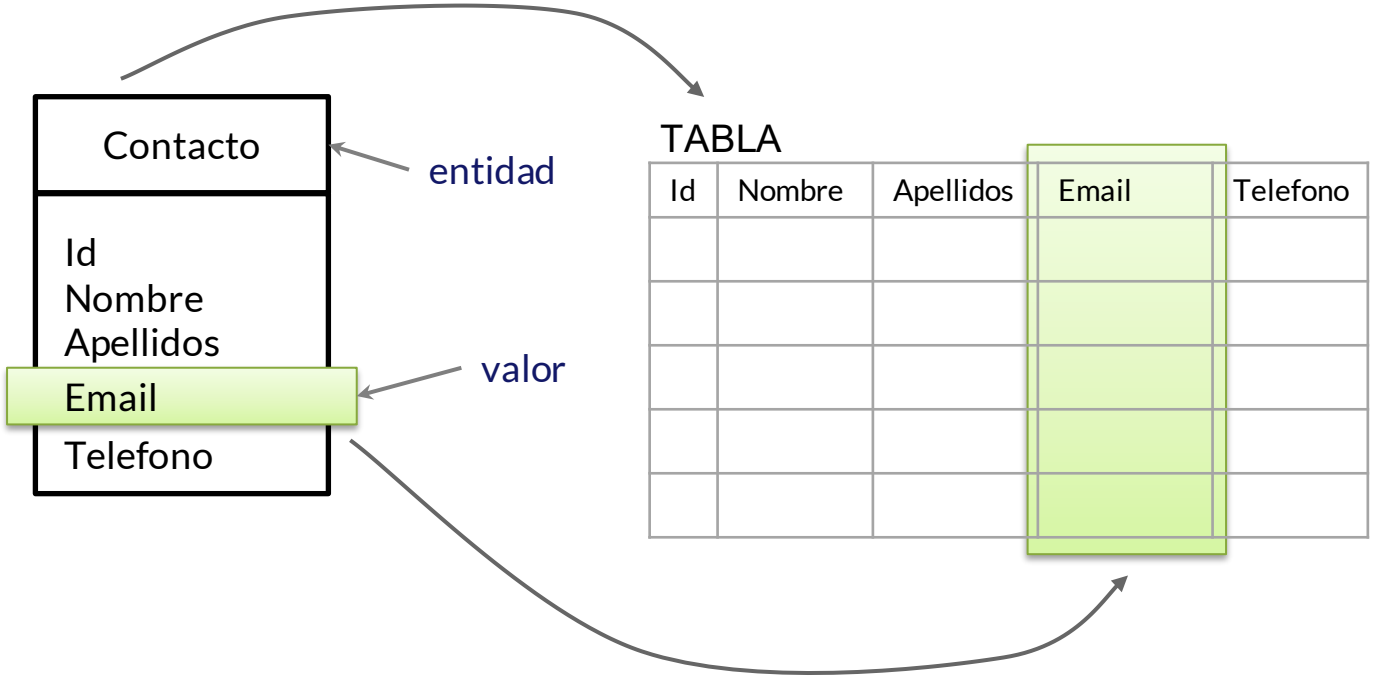
SPRING DATA JPA
ENTIDADES

MODELO DE DOMINIO
DE UN SISTEMA



- Representa los conceptos propios del problema a resolver.
- Vocabulario y conceptos clave.
- **Entidades** y **relaciones**.

MAPEO DE OBJETOS A TABLAS



ENTIDAD

- Clase (*pojo*) Java con *@Entity*
- Debe tener (al menos) un atributo que lo identifique (concepto de PK de bases de datos) con @Id. Anotación sobre propiedad o método *getter*.
- También se trasladarán a la base de datos como columnas el resto de propiedades que tenga la clase.

ENTIDAD

```
@Entity
public class Producto {

        @Id @GeneratedValue
        private Long Id;

        private String nombre;
        private String descripcion;
        private float pvp;

        //… resto de métodos y atributos
}
```

CONTROL DE NOMBRES

- *@Entity* se mapea con una tabla que se llame igual que la clase.
- Con un *@Table* adicional, podemos cambiar el nombre.

```
@Entity
@Table(name="PROD")
public class Producto {

        //… resto de métodos y atributos
}
```

CONTROL DE NOMBRES

- Los atributos también se mapean con su nombre
- Con un *@Column*, podemos cambiar el nombre.

```
@Entity
@Table(name="PROD")
public class Producto {

        @Column(name="prod_nombre")
    private String nombre;
        //… resto de métodos y atributos
}
```

ANOTACIÓN @Column

- Además *@Column* nos permite definir otras propiedades:
  - *Nullable:* si permite almacenar nulos
  - *Name*: nombre de la columna en la BD
  - *Insertable, updatable:* define si la entidad puede ser o no insertable o actualizable.
  - *Length:* tamaño que tendrá el campo en la BD.

ELECCIÓN DEL IDENTIFICADOR (CLAVE PRIMARIA)

- Es habitual utilizar campos *artificiales*, con el mismo nombre (*id)* y de tipo entero (*long*).
- JPA se puede encargar de generarlo (*@GeneratedValue*).
  - **AUTO: Hibernate escoge la mejor estrategia para el SGBD elegido**.
  - SEQUENCE: se utiliza una secuencia
  - IDENTITY: se utiliza un campo autonumérico
  - TABLE: se utiliza una tabla extra especial.

MAPEO DE VALORES.
TIPOS DE DATOS

- Tipos básicos
- Envoltorios de tipos básicos (***Long***, *Double, …)*
- ***String***, *BigInteger*, *BigDecimal*, *java.util.Date*, *java.util.Calendar*, *java.sql.Date*, *java.sql.Time*, *java.sql.Timestamp*, *byte[]*, *Byte[]*, *char[]*, o *Character[]*
- *java.io.Serializable* (representación en bytes)*
- *@Embeddable*

El resto de tipos de datos generarán por defecto un error.

MAPEO DE VALORES. ASOCIACIONES

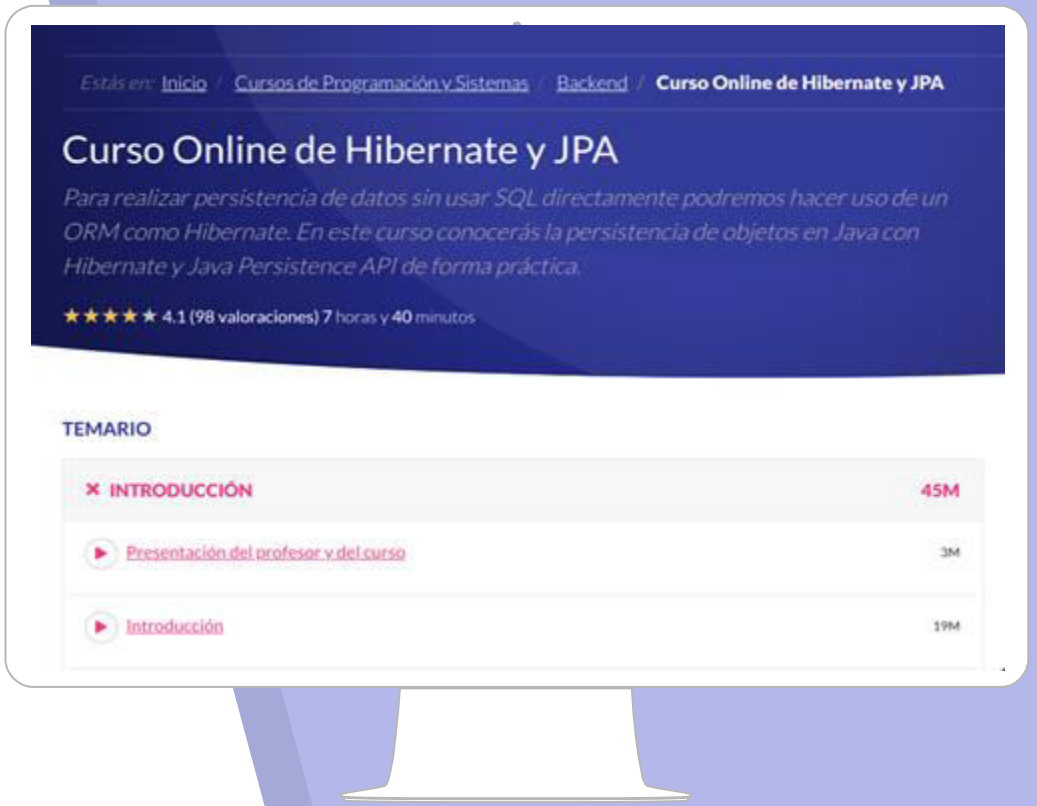
- JPA nos permite asociar dos entidades.
- Debemos conocer la multiplicidad de dicha asociación.
  - @ManyToOne: muchos a uno
  - @OneToMany: uno a muchos
  - @ManyToMany: muchos a muchos
  - @OneToOne: uno a uno.

MAPEO DE VALORES. ASOCIACIONES

```
@Entity
public class Producto {

        @Id @GeneratedValue
        private Long Id;

        //… resto de atributos

        @ManyToOne
        private Categoria categoria;

}
```

ENTIDADES
JPA

Accede a nuestro Curso de Hibernate y JPA para saber más sobre entidades y asociaciones



EN NUESTRO PROYECTO

- Vamos a transformar nuestra clase modelo en una **entidad.**

```
@Entity
public class Empleado {

    @Id @GeneratedValue
    @Min(value=0, message="{empleado.id.mayorquecero}")
    private long id;
```

- Escogemos siempre las anotaciones de *javax.persistence.**, y no las específicas de Hibernate.

EN NUESTRO PROYECTO

- Para ver resultados, necesitamos esperar a la siguiente lección.