# **PROYECTO FINAL 1ºDAM Guía de etiquetas y beans**

***1 Spring Data JPA***

**@Entity:** se mapea con una tabla que se llama igual que la clase.

**@Table:** si lo escribimos debajo de *“@Entity”* podremos cambiar el nombre de esa tabla.

*@Entity*

*@Table(name=”PROD”)*

*public class Producto {*

*//… resto de métodos y atributos*

*}*

**@Column:** permite cambiar varias características de una columna.

**@Id:** establece la clave primaria de la tabla.

Para elegir la clave primaria es habitual utilizar campos artificiales, con el mismo nombre (id) y de tipo entero (long).

JPA se puede encargar de generarlo (**@GeneratedValue**).

**@Builder:** permite crear constructores. Sin *“@AllArgsConstructor”* este bean no funciona.

COMPOSICIÓN

@Cascade:

@orphanRemoval:

**@CreateDate:** crea la fecha en el momento de hacer una acción.

**.requestMatchers(“/admin/\*\*”).hasRole(“ADMIN”)**

@ManyToOne

private Categoria categoria;

Public String nuevoProd (Model model){

mode.addAtribute(“producto”, new Producto());

}

<option th:each=”categoria : ${categorias}”

th:value=”

<th:if”${param.error}

public List<Producto> generarProductosAAleatoriamente(int numero){

List<Long> listaIds = repo.obtenerIds();

Collections.shuffle(listaIds);

listaIds = listaIds.stream().limit(numero).collect(Collectors.toList());

return repo.findAllNyId(listaIds);

}

Meter @Lob a imagen y descripción.