

JPA @Table y @Entity son dos anotaciones de JPA que a veces nos generan dudas ya que nos hacen pensar que son muy parecidas e incluso tenemos dudas de si hacen falta las dos . Vamos a hablar un poco de ellas . En primer lugar @Table es la anotación que nos permite mapear nuestra entidad contra una tabla que tenga un nombre distinto. Es decir si por ejemplo tenemos la siguiente entidad.

```
package com.arquitecturajava.aplicacion.adapters.out.jpa;
```

```
import javax.persistence.Entity;  
import javax.persistence.Id;  
import javax.persistence.JoinColumn;  
import javax.persistence.ManyToOne;  
import javax.persistence.Table;
```

```
@Entity(name="Producto")  
@Table(name="Productos")  
public class ProductoEntity {  
  
    @Id  
    private String numero;  
    private String concepto;  
    private double importe;  
  
    @ManyToOne  
    @JoinColumn(name = "categoria_id")  
    private CategoriaEntity categoria;  
  
    public CategoriaEntity getCategoria() {  
        return categoria;  
    }  
}
```

```
public void setCategoria(CategoriaEntity categoria) {
    this.categoria = categoria;
}

public String getNumero() {
    return numero;
}

public void setNumero(String id) {
    this.numero = id;
}

public String getConcepto() {
    return concepto;
}

public void setConcepto(String concepto) {
    this.concepto = concepto;
}

public double getImporte() {
    return importe;
}

public void setImporte(double importe) {
    this.importe = importe;
}

public ProductoEntity(String numero, String concepto, double
importe) {
    super();
}
```

```

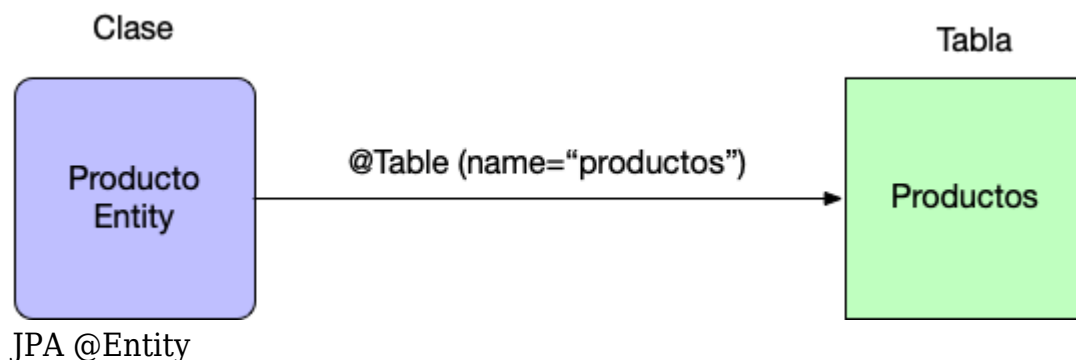
        this.numero = numero;
        this.concepto = concepto;
        this.importe = importe;
    }

    public ProductoEntity() {
        super();
    }
}

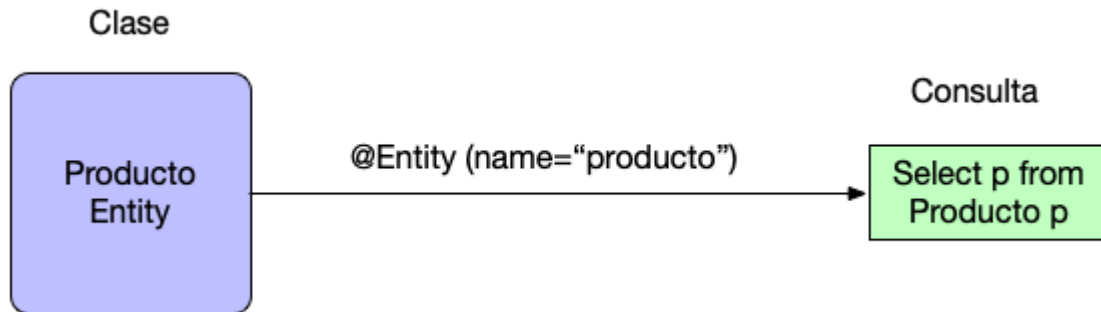
```

JPA @Table

Como podemos observar estamos utilizando @Table . Esta anotación nos permite decidir contra que tabla de la base de datos nuestra entidad se va a mapear . En muchas situaciones la entidad no tiene el mismo nombre exacto de la base de datos y nos viene bien usar esta anotación.



En muchas ocasiones es un poco más difícil de entender para que se usa @Entity con un name ya que @Entity simplemente registra la entidad. Esto se hace porque hay situaciones en las cuales a nivel de JPA Query Language queremos que el nombre de la entidad a nivel de consulta no sea idéntico al nombre de la clase para permitir simplificar a nivel de consulta



Tengamos siempre en cuenta estas dos anotaciones JPA @Table y JPA @Entity cuando trabajamos con JPA ya que en muchos casos su uso es prácticamente obligatorio ya sea una u otra.

- [JPA Query Language Objetos vs Tablas](#)
- [HTTP Response Status Codes y su uso](#)
- [Eclipse JPA y clases de dominio](#)
- [Anotaciones Java \(II\)](#)
- [¿DTO vs Entity? en Domain Driven Design](#)
- [JPA](#)