

Anotaciones JPA

@Entity entidad

@Table (name = "nombre Tabla") tabla

@Id atributo que es clave primaria

@GeneratedValue(strategy=GenerationType.IDENTITY) SEQUENCE, TABLE, AUTO

@Embedded

@Embeddable

@EmbeddedId para el atributo clave compuesta → @Embeddable la clase del atributo

@Column para cada atributo de la clase

- **name:** Permite establecer el nombre de la columna de la tabla con la que el atributo debe de mapear. A pesar de que ningún atributo de @Column es obligatorio, este atributo siempre recomiendo establecerlo.
- **length:** Permite definir la longitud de la columna en caracteres, solo aplica para Strings, en los demás tipos de datos será omitida. Solo se utiliza cuando JPA creará las tablas.
- **insertable:** Le indica a JPA si esa columna debe ser tomada en cuenta en los inserts, en caso de ser true, el valor será insertado, en caso contrario el valor será omitido y será colocado el valor default de la columna o null.
- **updatable:** Parecido al caso anterior, solo que en este caso la columna se toma en cuenta para operaciones de Update.
- **nullable:** Crea una restricción en la tabla (Not Null) para impedir que se inserten valores nulos. Solo se utiliza cuando JPA creará las tablas.
- **precisión:**
- **scale:** Utilizada únicamente para columnas que deben tener decimales, como valore recibe el número de decimales. Solo se utiliza cuando JPA creará las tablas.
- **unique:** Crea una restricción en la tabla para que sea el valor de esa columna sea única. Solo se utiliza cuando JPA creará las tablas.

@Enumerated

- String: permite persistir la enumeración por su nombre, lo que significa que será una columna alfanumérica. La anotación quedaría así:
@Enumerated(value = EnumType.STRING)
- Ordinal: esta estrategia persiste un valor entero que corresponde al valor ordinal o posición de valor en la enumeración. La anotación quedaría de la siguiente manera:
@Enumerated(value = EnumType.ORDINAL)

@Transient : evitar que se guarde y se lea de la BBDD

Relaciones

@OneToOne

@PrimaryKeyJoinColumn

@JoinColumn

@OneToMany

@JoinColumn

@ManyToOne

```
@ManyToMany
// Tabla relación libros autores
@JoinTable(
    name = "REL_LIBROS_AUTORES",
    joinColumns = {@JoinColumn(name = "FK_BOOK", nullable = false)},
    inverseJoinColumns = {@JoinColumn(name="FK_AUTHOR", nullable = false)}
)
@ManyToMany(mappedBy = "autores",fetch=FetchType.EAGER)
```

Herencia

```
@Inheritance(strategy=InheritanceType.SINGLE_TABLE)
@DiscriminatorValue(value="...")
@Inheritance(strategy=InheritanceType.JOINED)
@Inheritance(strategy=InheritanceType.TABLE_PER_CLASS)
```