

# ACTIVIDAD 3

## JPA

Adrián Caballero

David Lara

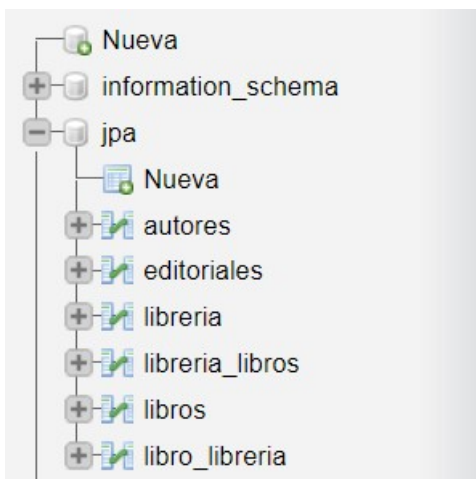
Hemos tenido problemas con el programa ya que al iniciarlo nos salen advertencias en la aplicación:

```
feb. 19, 2022 7:29:22 P. M. org.hibernate.jpa.internal.util.LogHelper logPersistenceUnitInformation
INFO: HHH000204: Processing PersistenceUnitInfo [name: JPA]
feb. 19, 2022 7:29:22 P. M. org.hibernate.Version logVersion
INFO: HHH000412: Hibernate ORM core version 5.4.21.Final
feb. 19, 2022 7:29:23 P. M. org.hibernate.annotations.common.reflection.java.JavaReflectionManager <clinit>
INFO: HCANN000001: Hibernate Commons Annotations {5.1.0.Final}
feb. 19, 2022 7:29:23 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl configure
WARN: HHH10001002: Using Hibernate built-in connection pool (not for production use!)
feb. 19, 2022 7:29:23 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001005: using driver [com.mysql.cj.jdbc.Driver] at URL [jdbc:mysql://localhost:3306/JPA?createDatabaseIfNotExist=true]
feb. 19, 2022 7:29:23 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001001: Connection properties: {user=root}
feb. 19, 2022 7:29:23 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl buildCreator
INFO: HHH10001003: Autocommit mode: false
feb. 19, 2022 7:29:23 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PooledConnections <init>
INFO: HHH000115: Hibernate connection pool size: 20 (min=1)
feb. 19, 2022 7:29:23 P. M. org.hibernate.dialect.Dialect <init>
INFO: HHH000400: Using dialect: org.hibernate.dialect.MySQL8Dialect
feb. 19, 2022 7:29:24 P. M. org.hibernate.resource.transaction.backend.jdbc.internal.DdlTransactionIsolatorNonJtaImpl getIsolatedConnection
INFO: HHH10001501: Connection obtained from JdbcConnectionAccess [org.hibernate.engine.jdbc.env.internal.JdbcEnvironmentInitiator$ConnectionProviderJdbcConnectionAccess@780]
feb. 19, 2022 7:29:24 P. M. org.hibernate.engine.transaction.jta.platform.internal.JtaPlatformInitiator initiateService
INFO: HHH000490: Using JtaPlatform implementation: [org.hibernate.engine.transaction.jta.platform.internal.NoJtaPlatform]
feb. 19, 2022 7:29:24 P. M. org.hibernate.engine.jdbc.connections.internal.DriverManagerConnectionProviderImpl$PoolState stop
INFO: HHH10001008: Cleaning up connection pool [jdbc:mysql://localhost:3306/JPA?createDatabaseIfNotExist=true]
```

Sinceramente no hemos encontrado solución a esos problemas.

Aunque el programa funciona.

BBDD creada:



1. Autor, tendrá un id, un nombre, unos apellidos y una fecha de nacimiento. Un autor podrá escribir muchos libros

<div><div><div>←</div><div>T</div><div>→</div></div></div>						id_autor	apellidos	fecha_Nacimiento	nombre
<div><div><div><div></div></div></div><div><div><div></div></div></div></div>	<div><div><div></div></div></div> Editar	<div><div><div></div></div></div> Copiar	<div><div><div></div></div></div> Borrar	1	CALLE MORENO 33, 2 A, LEGANES	22-3-45	AUTOR-1		
<div><div><div><div></div></div></div><div><div><div></div></div></div></div>	<div><div><div></div></div></div> Editar	<div><div><div></div></div></div> Copiar	<div><div><div></div></div></div> Borrar	2	CALLE SIERRA MORENA 99	10-2-45	AUTOR-3		
<div><div><div><div></div></div></div><div><div><div></div></div></div></div>	<div><div><div></div></div></div> Editar	<div><div><div></div></div></div> Copiar	<div><div><div></div></div></div> Borrar	3	CALLE ALCALA 66, 1 B	12-2-32	AUTOR-2		









2. Editorial, tendrá un id, un nombre y una dirección. También tendrá una colección de libros publicados por la editorial.

						id_editorial	direccion	nombre	
<input type="checkbox"/>		Editar		Copiar		Borrar	1	CALLE MADRID 45, 3A	EDITORIAL 1
<input type="checkbox"/>		Editar		Copiar		Borrar	2	CALLE OROPESA 55	EDITORIAL 2

3. Libro, tendrá un id, un título, un precio, una editorial y un autor.

  				id_libro	precio	titulo	fk_id_autor	fk_id_editorial
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	34	libro6	3	1
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	29	libro4	3	1
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	3	77	libro8	1	1
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	4	8	libro3	3	2
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	5	9	libro5	3	2
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	6	89	libro7	3	2
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	7	2	libro2	1	1
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	8	55	libro1	2	2

4. Librería, tendrá un id, un nombre, un nombre del dueño, una dirección y una colección de libros. Además, hay que tener en cuenta que un libro puede estar en diferentes librerías.

  				id_libreria	direccion	dueño	nombre
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	1	PLAZA INDEPENDENCIA KIEV, NUMERO 1	ZELESNKI	libreria2
<input type="checkbox"/>	 Editar	 Copiar	 Borrar	2	PLAZA ROJA, MOSCU NUMERO 1	PUTIN	libreria1

Libreria_id_libreria	coleccion_id_libro
1	2
1	3
1	8
1	7
2	8
2	7
2	4
2	2

fk_id_libro	fk_id_libreria
1	1
1	2
2	2
3	1
3	2
4	2
5	1
5	2
6	1
6	2
7	2
8	2

Todas estas entidades tendrán relaciones bidireccionales.

```
//LIBRO1
```

```
libro1.setEditorial(editorial2);  
autor1.añadirLibro(libro1);  
libro1.setAutor(autor2);  
editorial2.añadirLibro(libro1);
```

```
//LIBRO2
```

```
libro2.setAutor(autor1);  
libro2.setEditorial(editorial1);  
autor2.añadirLibro(libro2);  
editorial2.añadirLibro(libro2);
```

```
//LIBRO3
```

```
libro3.setAutor(autor3);  
libro3.setEditorial(editorial2);  
autor3.añadirLibro(libro3);  
editorial1.añadirLibro(libro3);
```

```
//LIBRO4
```

```
libro4.setAutor(autor3);  
libro4.setEditorial(editorial1);  
autor1.añadirLibro(libro4);  
editorial2.añadirLibro(libro4);
```

```
//LIBRO5
```

```
libro5.setAutor(autor3);  
libro5.setEditorial(editorial2);  
autor2.añadirLibro(libro5);  
editorial1.añadirLibro(libro5);
```

```
//LIBRO6
```

```
libro6.setAutor(autor3);  
libro6.setEditorial(editorial1);  
autor3.añadirLibro(libro6);  
editorial2.añadirLibro(libro6);
```

```
//LIBRO7
```

```
libro7.setAutor(autor3);  
libro7.setEditorial(editorial2);  
autor1.añadirLibro(libro7);  
editorial1.añadirLibro(libro7);
```

Menú:

1. Mostrar todos los libros
2. Mostrar todos los autores
3. Mostrar todas las librerías
4. Mostrar todos los libros dados de alta
5. Salir

## 1- Mostrar todos los libros dados de alta, con su editorial y su autor

```
Titulo: libro6 - EDITORIAL 1, precio: 34.0, autor: AUTOR-3 CALLE SIERRA MORENA 99, editorial:
Titulo: libro4 - EDITORIAL 1, precio: 29.0, autor: AUTOR-3 CALLE SIERRA MORENA 99, editorial:
Titulo: libro8 - EDITORIAL 1, precio: 77.0, autor: AUTOR-1 CALLE MORENO 33, 2 A, LEGANES, editorial:
Titulo: libro3 - EDITORIAL 2, precio: 8.0, autor: AUTOR-3 CALLE SIERRA MORENA 99, editorial:
Titulo: libro5 - EDITORIAL 2, precio: 9.0, autor: AUTOR-3 CALLE SIERRA MORENA 99, editorial:
Titulo: libro7 - EDITORIAL 2, precio: 89.0, autor: AUTOR-3 CALLE SIERRA MORENA 99, editorial:
Titulo: libro2 - EDITORIAL 1, precio: 2.0, autor: AUTOR-1 CALLE MORENO 33, 2 A, LEGANES, editorial:
Titulo: libro1 - EDITORIAL 2, precio: 55.0, autor: AUTOR-2 CALLE ALCALA 66, 1 B , editorial:
```

## 2-Mostrar todos los autores dados de alta, con sus libros asociados

```
Autor: AUTOR-1 CALLE MORENO 33, 2 A, LEGANES
LIBROS DEL : AUTOR-1
Libro: libro8
Libro: libro2
Autor: AUTOR-2 CALLE ALCALA 66, 1 B
LIBROS DEL : AUTOR-2
Libro: libro1
Autor: AUTOR-3 CALLE SIERRA MORENA 99
LIBROS DEL : AUTOR-3
Libro: libro6
Libro: libro4
Libro: libro3
Libro: libro5
Libro: libro7
```

## 3. Mostrar todas las librerías, con solamente sus libros asociados

```
Libreria: libreria2
Libro: libro4
Libro: libro8
Libro: libro1
Libro: libro2
Libreria: libreria1
Libro: libro1
Libro: libro2
Libro: libro3
Libro: libro4
```



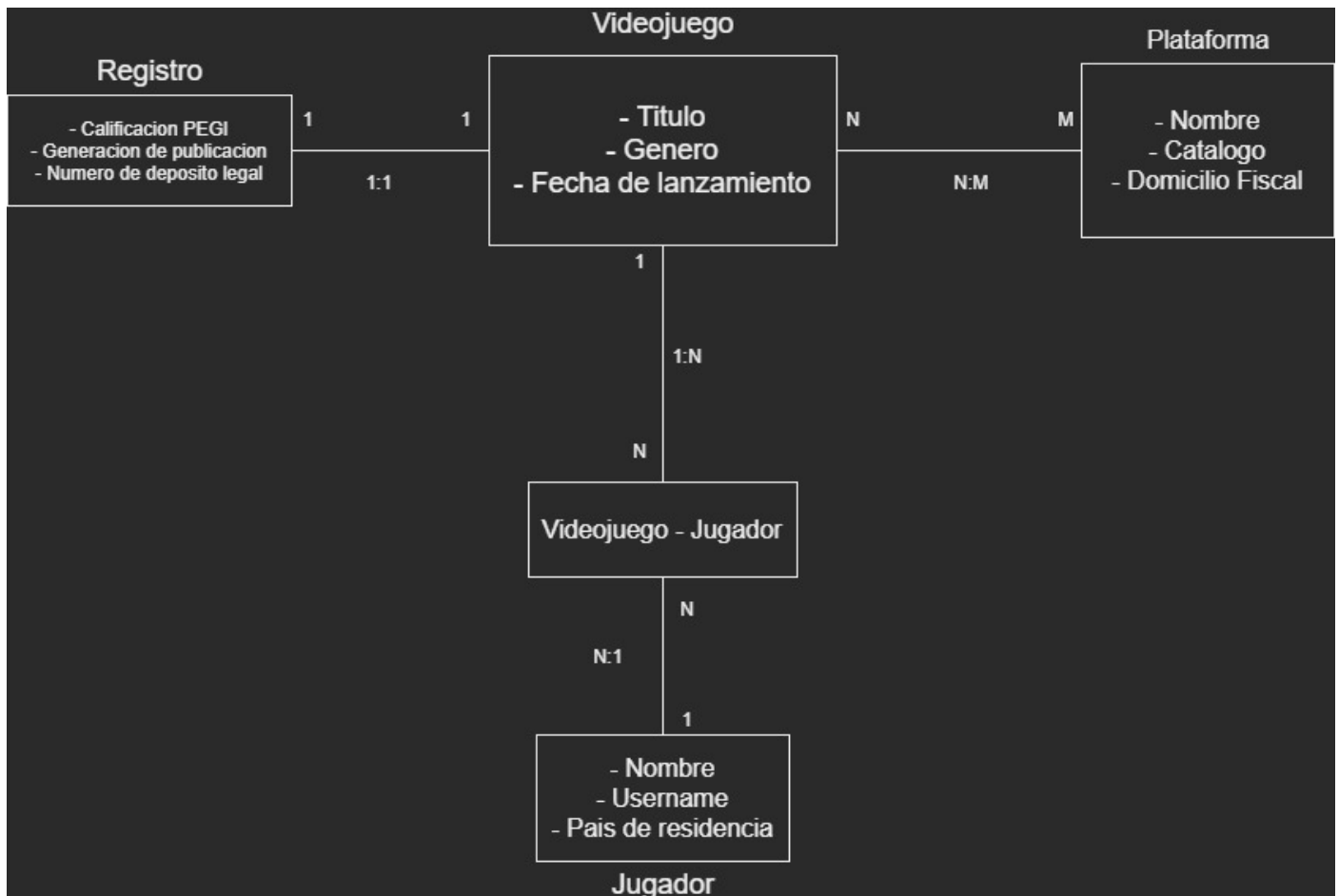
4. Mostrar todos los libros dados de alta, y en la librería en la que están.

```
libro6
  esta disponible en las siguientes librerías
librería2
librería1
libro4
  esta disponible en las siguientes librerías
librería1
libro8
  esta disponible en las siguientes librerías
librería2
librería1
libro3
  esta disponible en las siguientes librerías
librería1
libro5
  esta disponible en las siguientes librerías
librería2
librería1
libro7
  esta disponible en las siguientes librerías
librería2
librería1
libro2
  esta disponible en las siguientes librerías
librería1
libro1
  esta disponible en las siguientes librerías
librería1
```

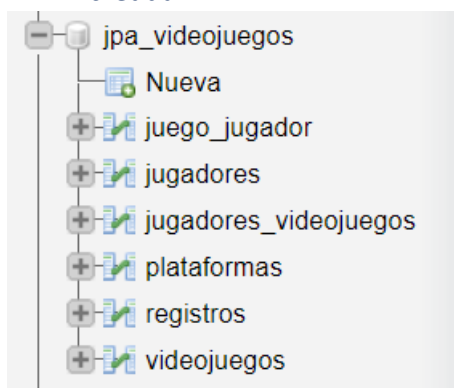
### Requerimiento 3

Se pide realizar al menos un modelo de datos (aunque se recomienda que cada integrante del equipo realice uno distinto) que cubra todos los tipos de relaciones visto en clase, esto es, uno a uno, uno a N y N a M. El modelo tendrá las entidades necesarias para realizar dichas relaciones. Solo se piden las entidades y las anotaciones en cada entidad para realizar el modelo de datos de JPA.

#### MODELO DE DATOS:



BBDD creada:



### Entidad Registro:

```
1 @Entity
2 @Table(name = "registros")
3 public class Registro {
4
5     @Id
6     @GeneratedValue(strategy = GenerationType.IDENTITY)
7     private int depositoLegal;
8
9     private String pegi;
10
11     private String publicacion;
12
13     @OneToMany(mappedBy = "registro", cascade = CascadeType.ALL)
14     private List<Videojuego> videojuegos;
15 }
```

### Entidad Videojuego:

```
17 @Entity
18 @Table(name="videojuegos")
19
20 public class Videojuego {
21
22     @Id
23     @GeneratedValue(strategy = GenerationType.IDENTITY)
24     private int id;
25
26     private String titulo;
27     private String genero;
28     private String fecha_Lanzamiento;
29
30     @ManyToOne (cascade = CascadeType.PERSIST)
31     @JoinColumn(name = "fk_id_plataforma", referencedColumnName = "id_plataforma")
32     private Plataforma plataforma;
33
34     @ManyToOne(cascade = CascadeType.PERSIST)
35     @JoinColumn(name = "fk_depositoLegal", referencedColumnName = "depositoLegal")
36     private Registro registro;
37
38     @ManyToMany(cascade=CascadeType.PERSIST)
39     @JoinTable(name="juego_jugador",
40         joinColumns = {@JoinColumn(name="fk_id", referencedColumnName="id")},
41         inverseJoinColumns = {@JoinColumn(name="fk_id_juegos", referencedColumnName="id_juegos")})
42 }
```

### Entidad Plataforma:

```
15 @Entity
16 @Table(name = "plataformas")
17
18 public class Plataforma {
19
20     @Id
21     @GeneratedValue(strategy = GenerationType.IDENTITY)
22     private int id_plataforma;
23
24     private String nombre;
25
26     private String catalogo;
27     private String direccion;
28
29     @OneToMany(mappedBy = "plataforma", cascade = CascadeType.ALL)
30     private List<Videojuego> juegosPublicados;
31
32     public Plataforma() {
33     }
```



Entidad Jugador:

```
15 @Entity
16 @Table(name = "jugadores")
17 public class Jugador {
18
19     @Id
20     @GeneratedValue(strategy = GenerationType.IDENTITY)
21     private int id_juegos;
22     private String nombre;
23     private String username;
24     private String pais_Residencia;
25
26     private String dueño;
27     @ManyToMany(cascade = CascadeType.PERSIST)
28     private List<Videojuego> coleccionJuegos;
29 }
```