

11^a
Emisión

DIPLOMADO Desarrollo de Sistemas con Tecnología Java

Módulo 7 Persistencia con Spring Data

Dr. Omar Mendoza González

omarmendoza564@aragon.unam.mx



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Dirección General de Cómputo y de Tecnologías de información y Comunicación
Dirección de Docencia en TIC



Educación
Continua
1971 - 2021

Relaciones y Consultas

- Tipos de relaciones
 - Many-to-One
 - Many-to-Many
 - One-to-Many
 - One-to-One
- Anotaciones
 - @ManyToOne
 - @ManyToMany
 - @OneToMany
 - @OneToOne

Relaciones

- @ManyToOne
 - Existe relación Many-To-One entre las entidades donde se hace referencia a una entidad (columna o conjunto de columnas) con valores únicos que contienen de otra entidad (columna o conjunto de columnas).
 - En BDR, estas relaciones se aplican mediante el uso de clave primaria y clave foránea entre las tablas.
 - Permite considerar un ejemplo de una relación entre entidades alumno y departamento de manera unidireccional,
 - Eso significa que cada registro de alumno contiene un id de departamento, que debe ser una clave principal en la tabla Departamento.

Relaciones

- @ManyToMany
 - Relación de M:N donde una o más filas de una entidad se asocian a más de una fila en otra entidad.
 - Considerar un ejemplo de una relación entre dos entidades: **clase** y **profesor**.
 - De manera bidireccional, tanto clase como profesor tienen relación 1:M.
 - Cada registro de clase es referido por el profesor (ids de profesor), que debe ser llave primaria en la tabla del profesor y viceversa.

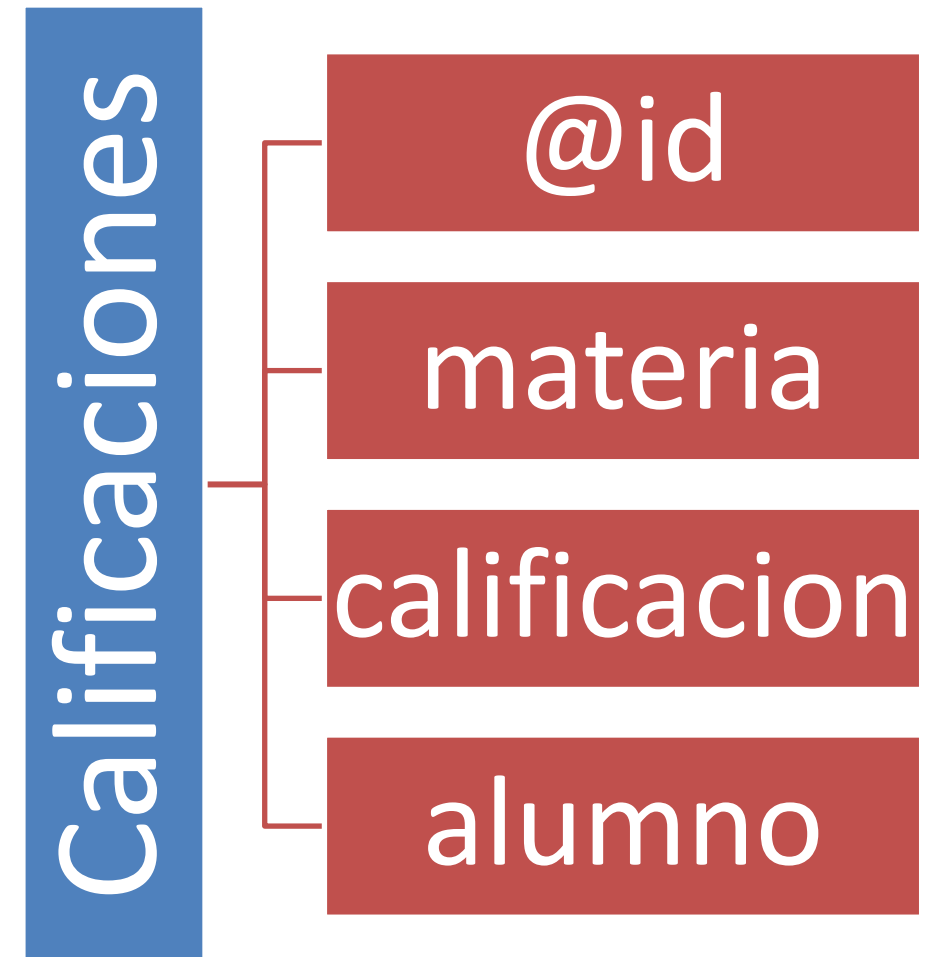
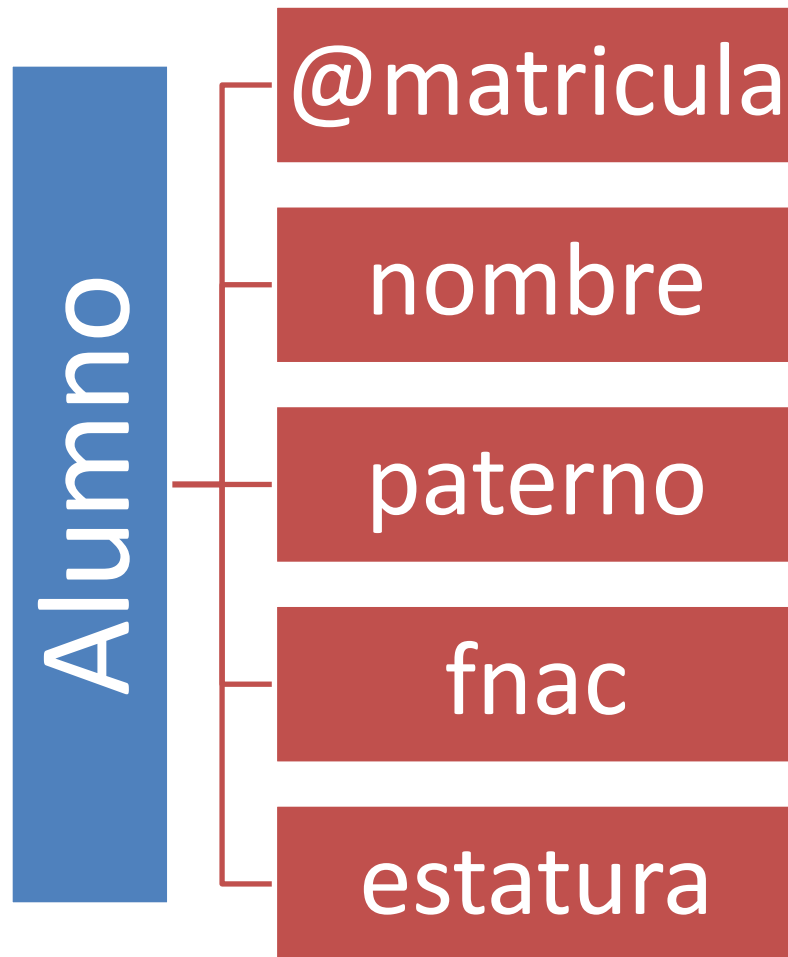
Relaciones

- @OneToMany
 - Cada fila de una entidad hace referencia a muchos registros secundarios en otra entidad.
 - Lo importante es que los registros secundarios no pueden tener varios padres.
 - En una relación uno a varios entre la tabla A y B de la tabla, cada fila en la tabla A puede ser vinculado a una o varias filas en la tabla B.
 - Considerar el ejemplo que un alumno se conectan de manera unidireccional con la tabla titulo.

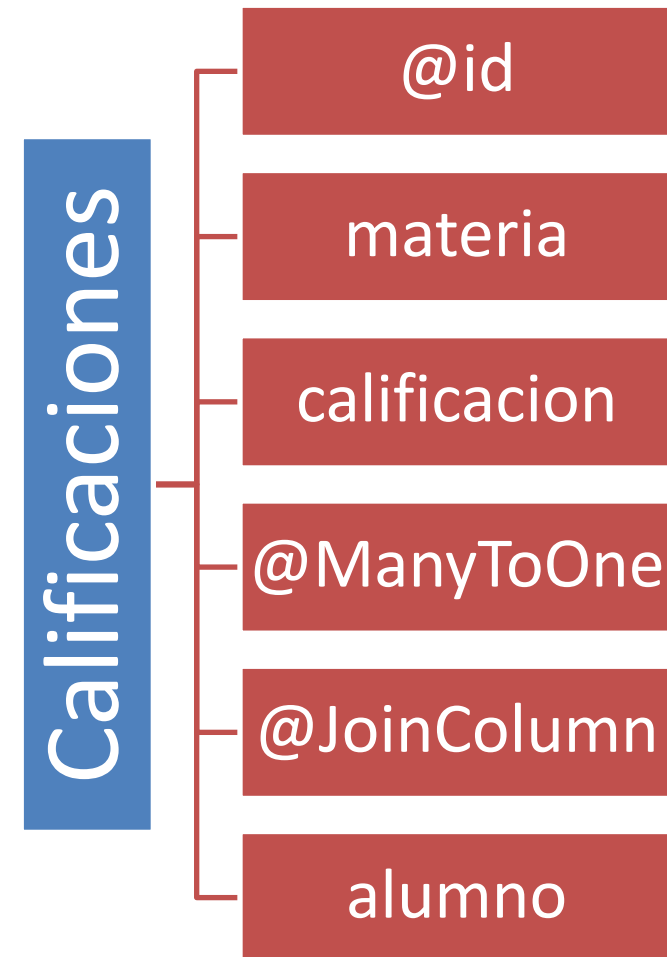
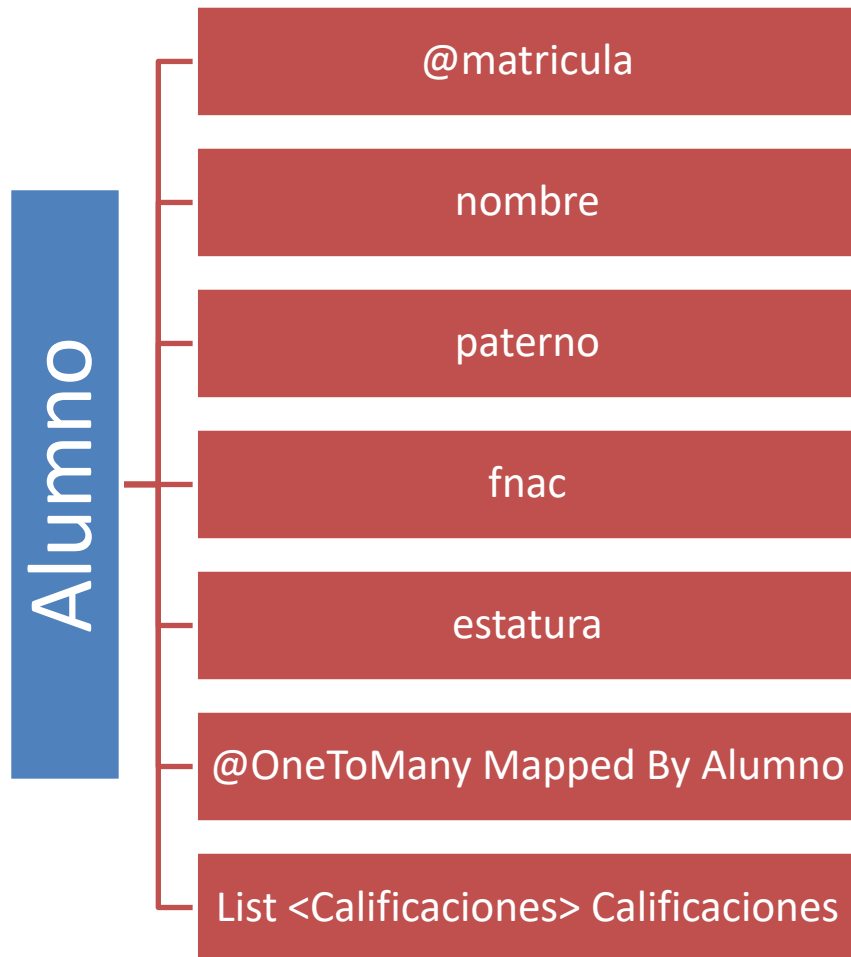
Relaciones

- @OneToOne
 - En una relación 1:1, un elemento puede vincularse a un único otro elemento.
 - Significa que cada fila de una entidad se refiere a una y sólo una fila de otra entidad.

Entidades



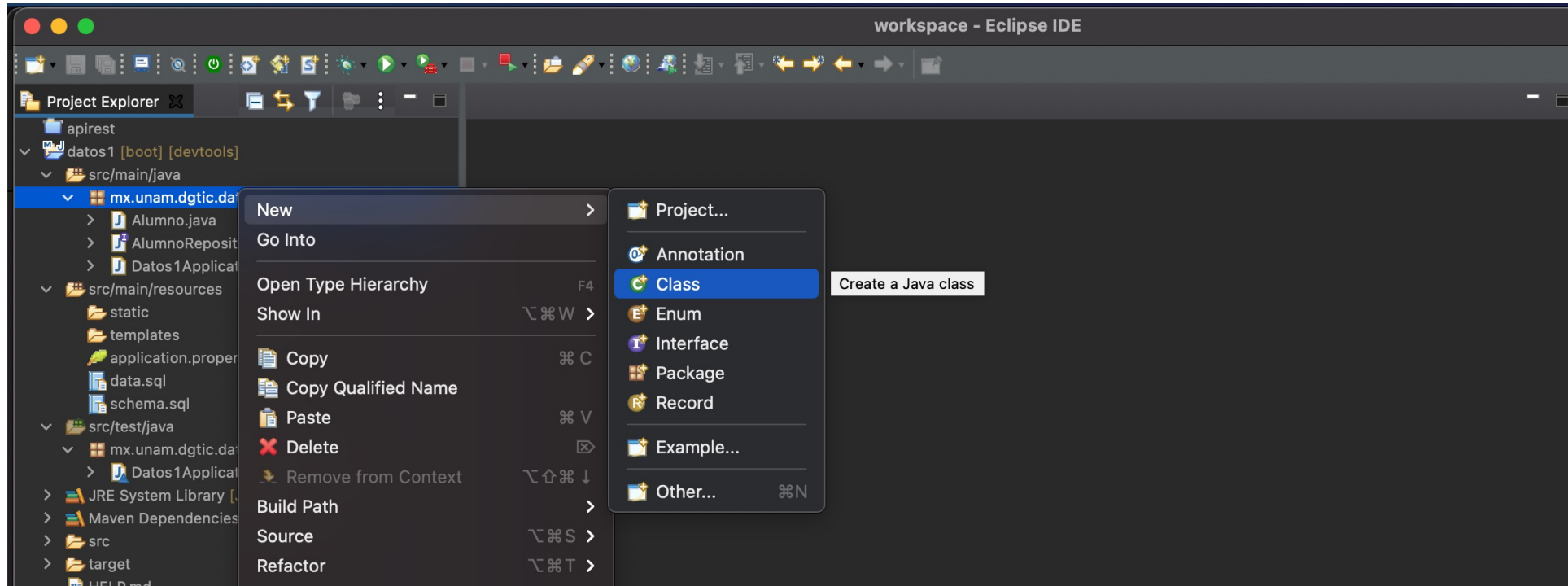
Relaciones



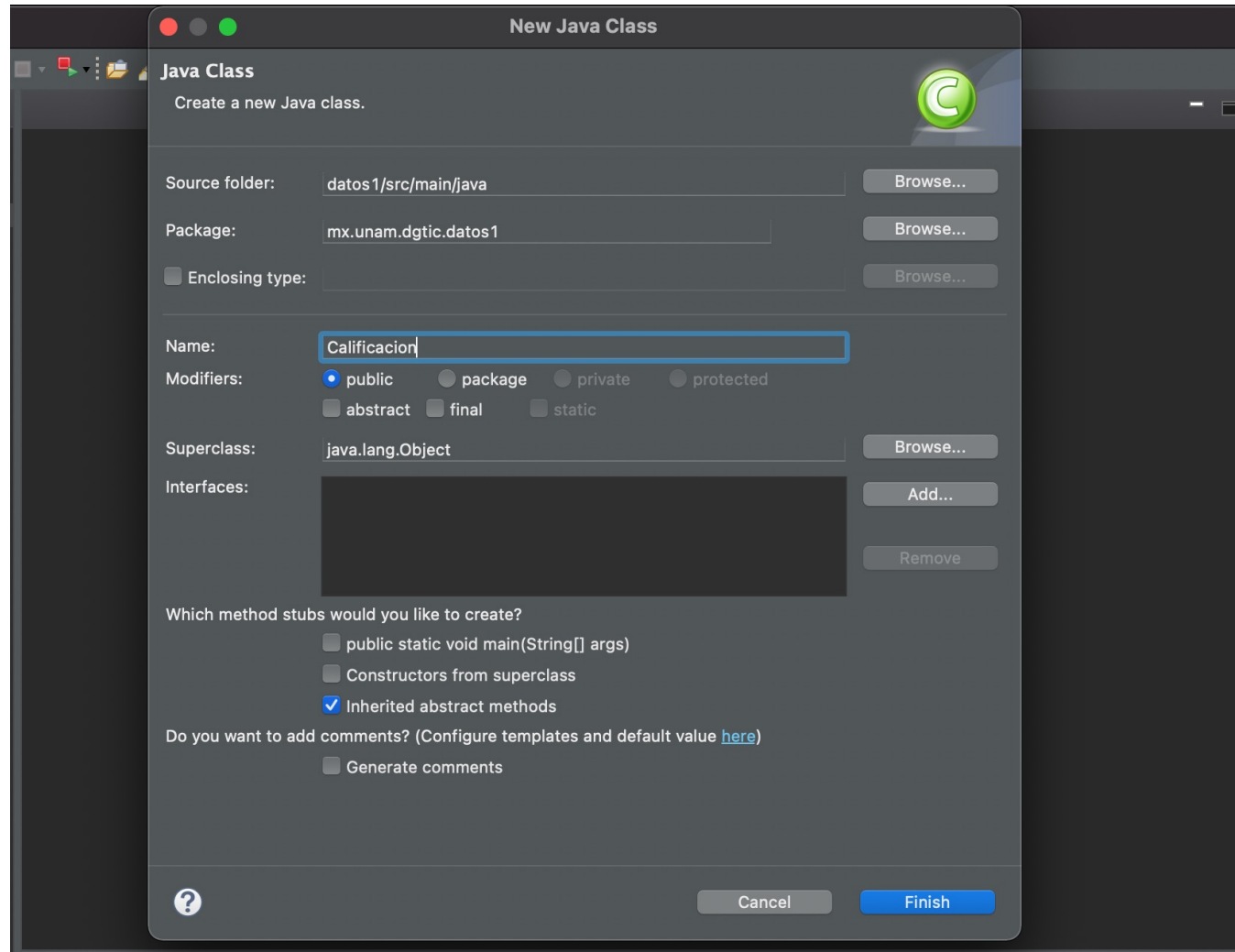
Relaciones



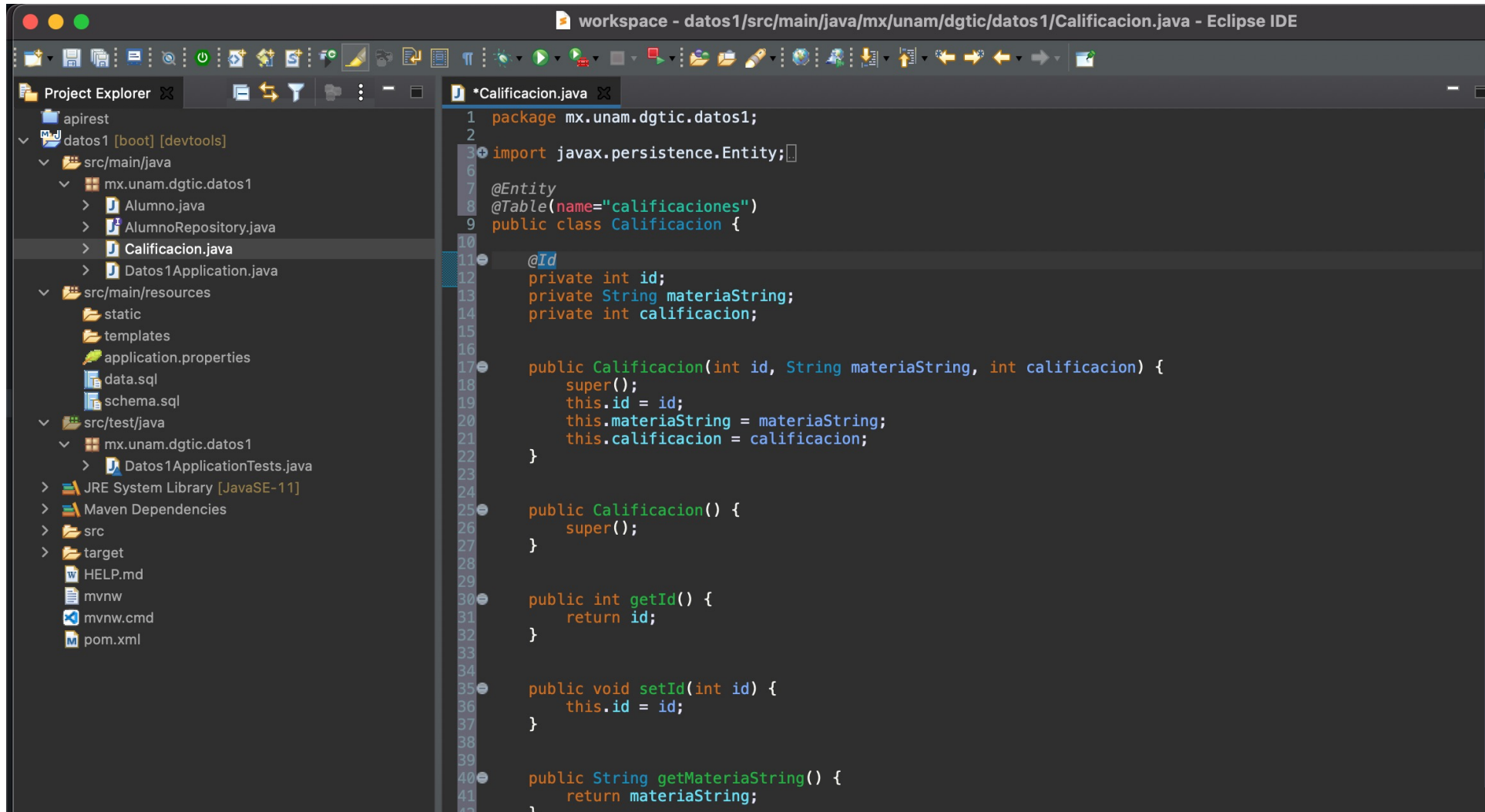
Entidad Calificacion



Entidad Calificacion



Entidad Calificacion



```
workspace - datos1/src/main/java/mx/unam/dgtic/datos1/Calificacion.java - Eclipse IDE

Project Explorer
  apirest
  datos1 [boot] [devtools]
    src/main/java
      mx.unam.dgtic.datos1
        Alumno.java
        AlumnoRepository.java
        Calificacion.java
        Datos1Application.java
    src/main/resources
      static
      templates
      application.properties
      data.sql
      schema.sql
    src/test/java
      mx.unam.dgtic.datos1
        Datos1ApplicationTests.java
  JRE System Library [JavaSE-11]
  Maven Dependencies
  src
  target
  HELP.md
  mvnw
  mvnw.cmd
  pom.xml

*Calificacion.java
1 package mx.unam.dgtic.datos1;
2
3 import javax.persistence.Entity;
4
5 @Entity
6 @Table(name="calificaciones")
7 public class Calificacion {
8
9     @Id
10     private int id;
11     private String materiaString;
12     private int calificacion;
13
14     public Calificacion(int id, String materiaString, int calificacion) {
15         super();
16         this.id = id;
17         this.materiaString = materiaString;
18         this.calificacion = calificacion;
19     }
20
21     public Calificacion() {
22         super();
23     }
24
25     public int getId() {
26         return id;
27     }
28
29     public void setId(int id) {
30         this.id = id;
31     }
32
33     public String getMateriaString() {
34         return materiaString;
35     }
36 }
```

Entidad Calificacion

Generar la relación con Alumno

```
//Generar la relacion M:1  
@ManyToOne  
@JoinColumn(name="alumnos_matricula")  
private Alumno alumno;
```

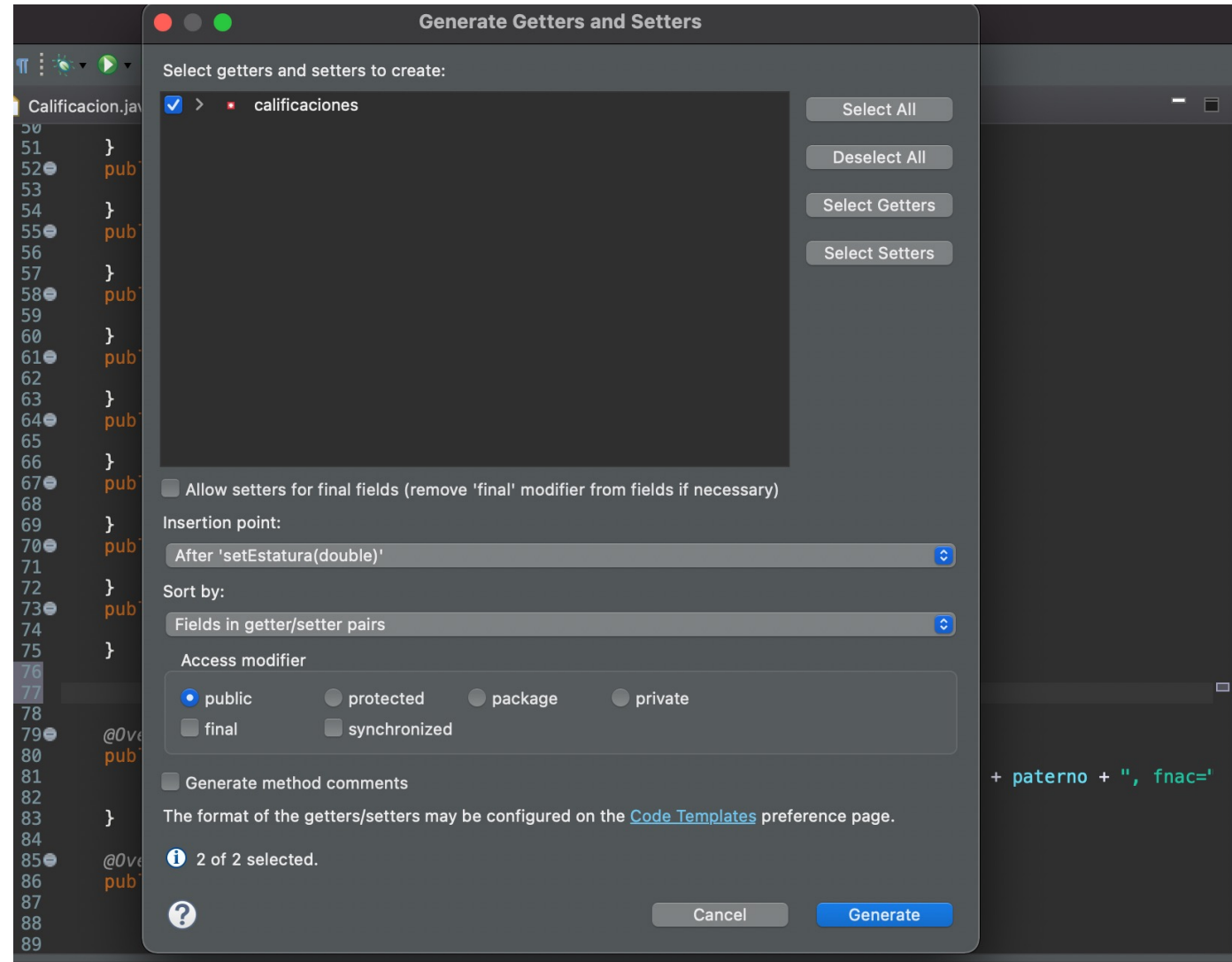
Entidad Alumno

Generar la relación con Calificacion

```
@OneToMany(mappedBy="alumno")  
private List<Calificacion> calificaciones = new ArrayList<Calificacion>();
```

Entidad Alumno

Agregar métodos Set y Get de calificaciones



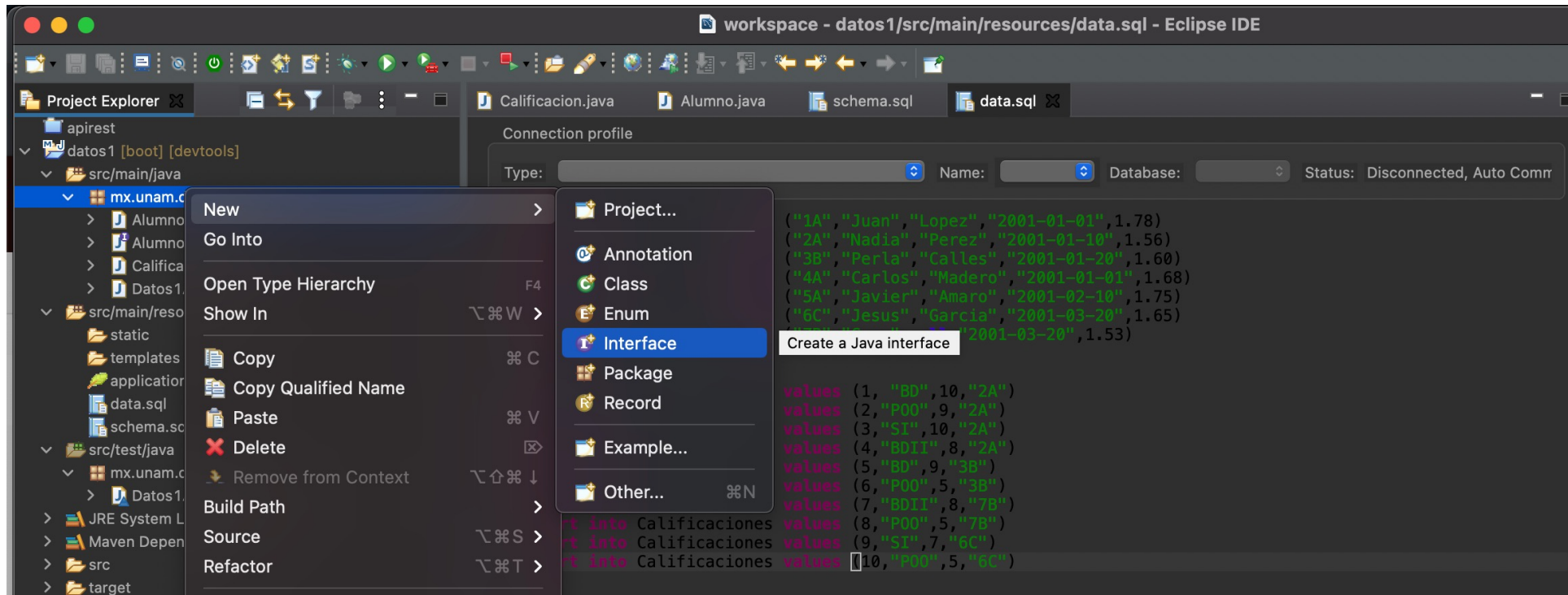
CalificacionRepository

CalificacionRepository

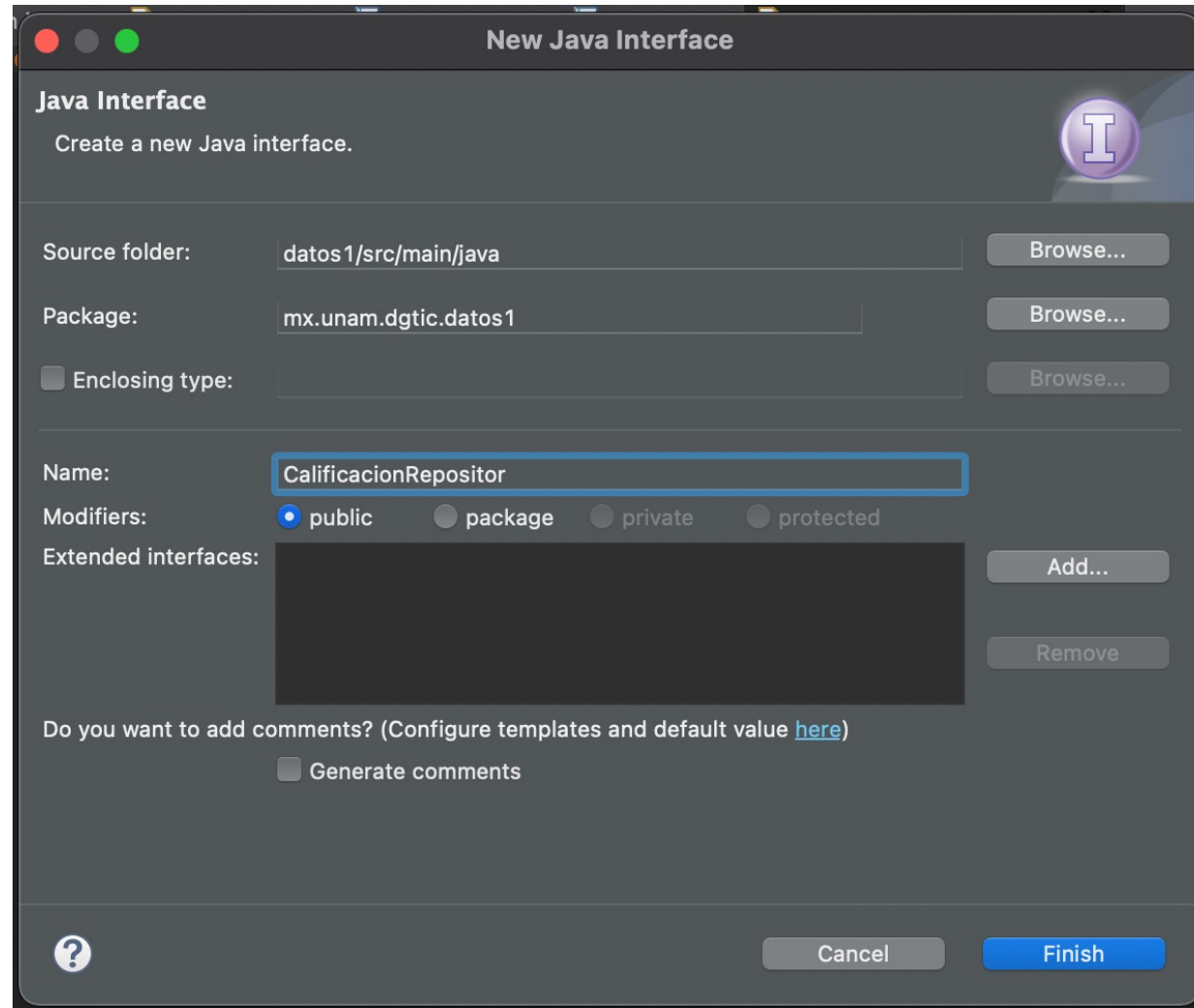
`findByMateria(String materia)`

`findByCalificacion(Int calificacion)`

CalificacionRepository



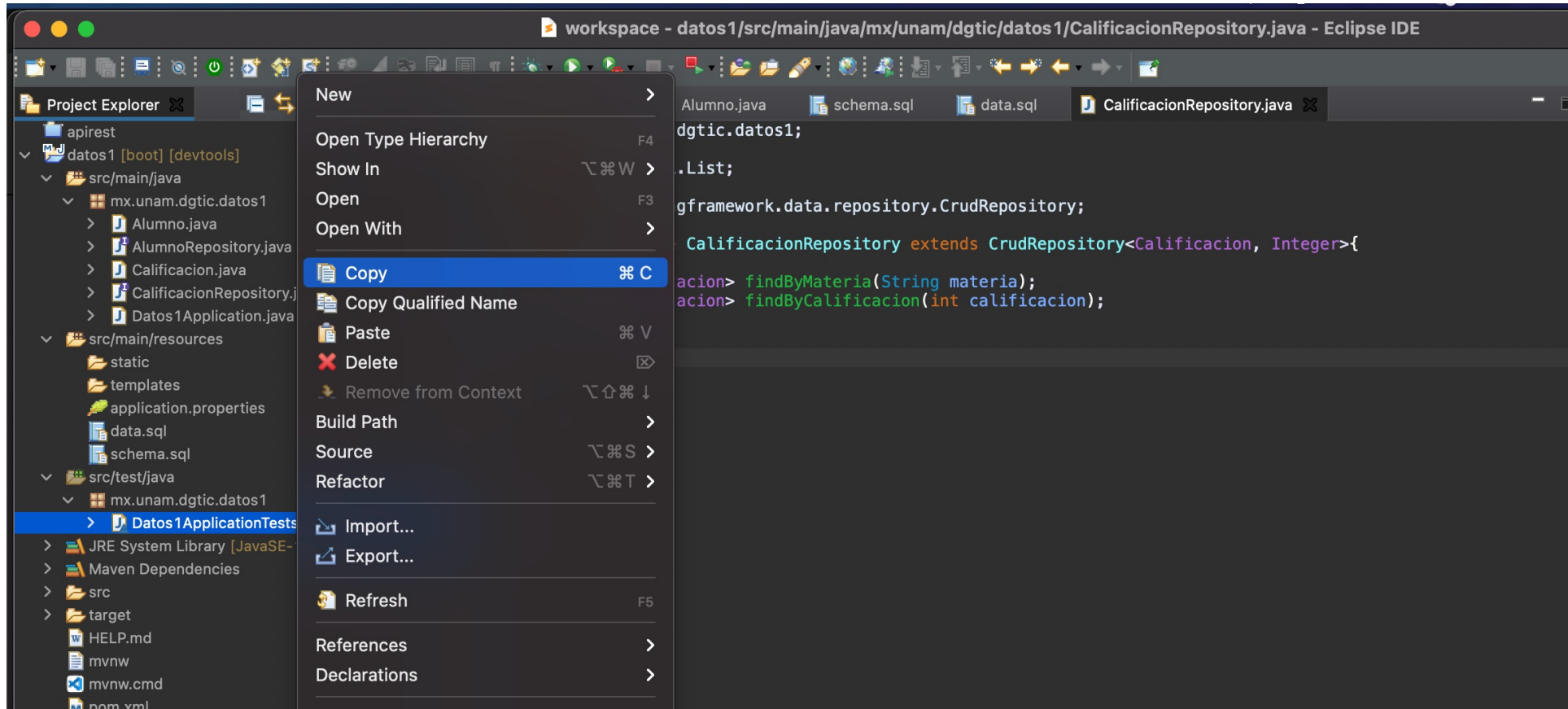
CalificacionRepository



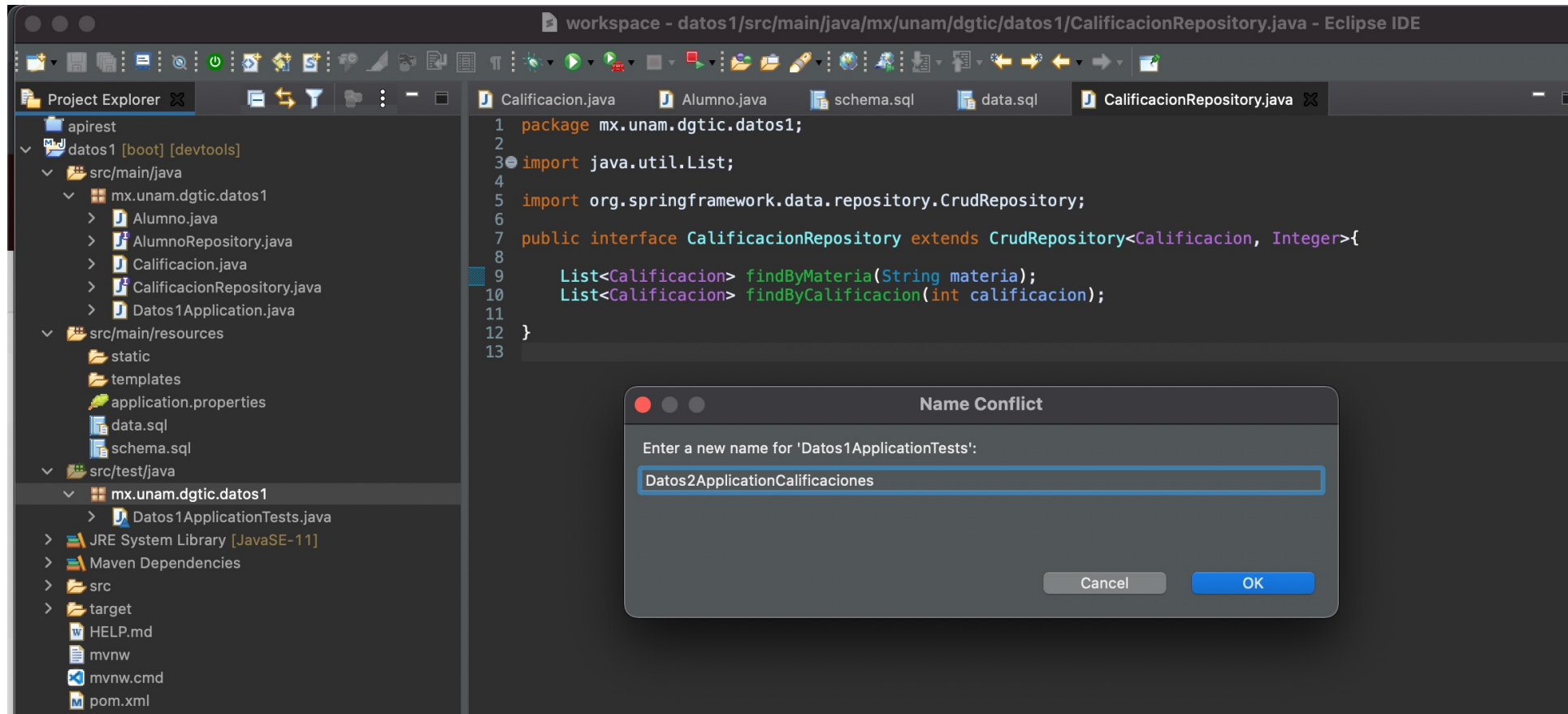
CalificacionRepository

```
1 package mx.unam.dgtic.datos1;
2
3 import java.util.List;
4
5 import org.springframework.data.repository.CrudRepository;
6
7 public interface CalificacionRepository extends CrudRepository<Calificacion, Integer>{
8
9     List<Calificacion> findByMateria(String materia);
10    List<Calificacion> findByCalificacion(int calificacion);
11
12 }
13
```

Nueva Prueba Unitaria



Nueva Prueba Unitaria



Nueva Prueba Unitaria

```
@SpringBootTest
//@Sql({ "/schema.sql", "/data.sql" })
class Datos2ApplicationCalificaciones {

    private static final String MATERIA = "BD";
    private static final double CALIFICACION = 8;

    @Autowired
    AlumnoRepository repositorioAlumno;

    @Autowired
    CalificacionRepository repositorioCalificacion;

    @Test
    void buscarPorTituloTest() {

        List<Calificacion> lista = repositorioCalificacion.findByMateria(MATERIA);

        System.out.println("findByMateria");
        lista.forEach(System.out::println);

        //assertThat(lista, hasItem(new Calificacion("BD")));
        assertThat(lista.size(), greaterThan(0));

    }

}
```


CalificacionRepository

CalificacionRepository

findByMateria(String materia)

findByCalificacion(Int calificacion)

findByAlumnoNombre(String nombre)

CalificacionRepository

```
//Buscar las calificaciones de un alumno  
List<Calificacion> findByAlumnoNombre(String nombre);
```

```
@Test  
void buscarPorAlumnoNombreTest() {  
  
    List<Calificacion> lista = repositorioCalificacion.findByAlumnoNombre("Nadia");  
  
    System.out.println("findByAlumnoNombre");  
    lista.forEach(System.out::println);  
  
    //assertThat(lista,hasItem(new Calificacion("BD")));  
    assertThat(lista.size(), greaterThan(0));  
  
}
```

```
findByAlumnoNombre  
Calificacion [id=1, materia=BD, calificacion=10, alumno=Alumno [matricula=2A, nombre=Nadia, paterno=Perez, fnac=2001-01-10 00:00:00.0, estatura=1.56]]  
Calificacion [id=2, materia=P00, calificacion=9, alumno=Alumno [matricula=2A, nombre=Nadia, paterno=Perez, fnac=2001-01-10 00:00:00.0, estatura=1.56]]  
Calificacion [id=3, materia=SI, calificacion=10, alumno=Alumno [matricula=2A, nombre=Nadia, paterno=Perez, fnac=2001-01-10 00:00:00.0, estatura=1.56]]  
Calificacion [id=4, materia=BDII, calificacion=8, alumno=Alumno [matricula=2A, nombre=Nadia, paterno=Perez, fnac=2001-01-10 00:00:00.0, estatura=1.56]]
```


Contacto

Dr. Omar Mendoza González

omarmendoza564@aragon.unam.mx