



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

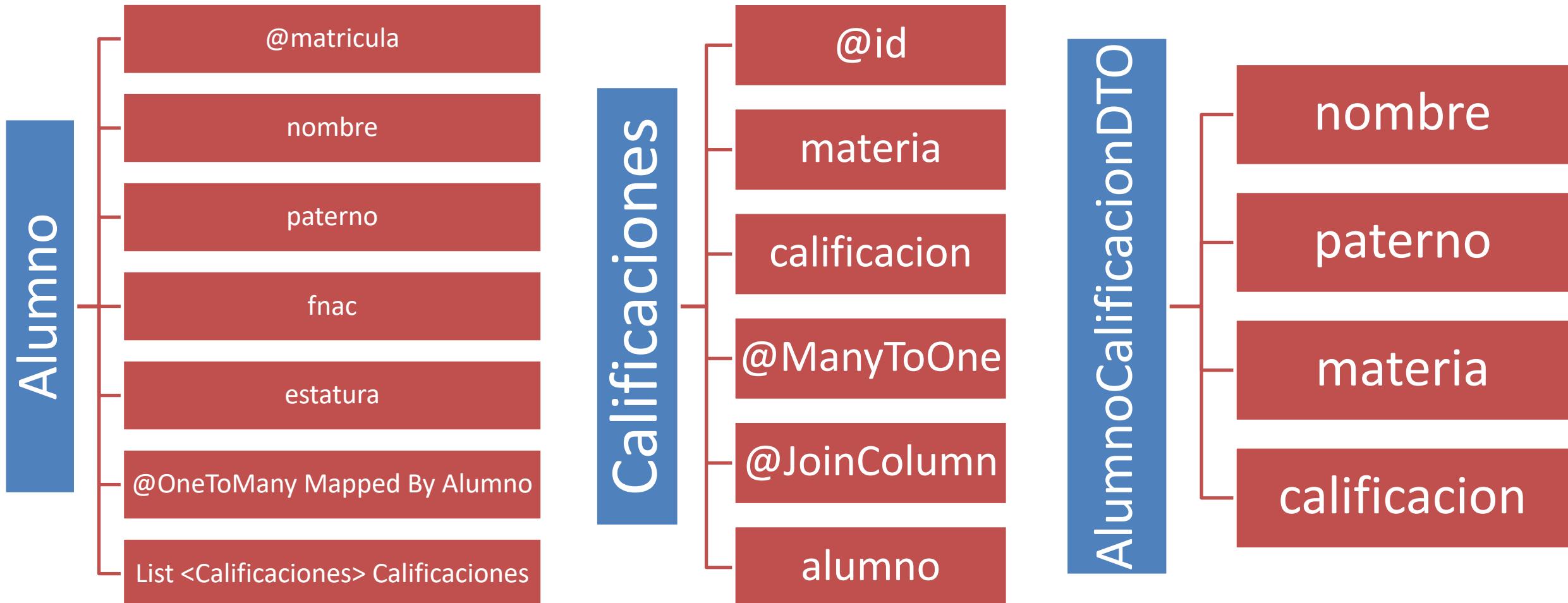


Educación
Continua
1971 - 2021

DTO

- DTOs (Data Transfer Objects)
- Ayuda a agrupar información de varias entidades generando una entidad propia
- El patrón DTO tiene como finalidad de crear un objeto plano (POJO) con una serie de atributos que puedan ser enviados o recuperados del servidor en **una sola invocación**
- De tal forma que un DTO puede contener información de **múltiples fuentes** o tablas y concentrarlas en una única clase simple.

TDO

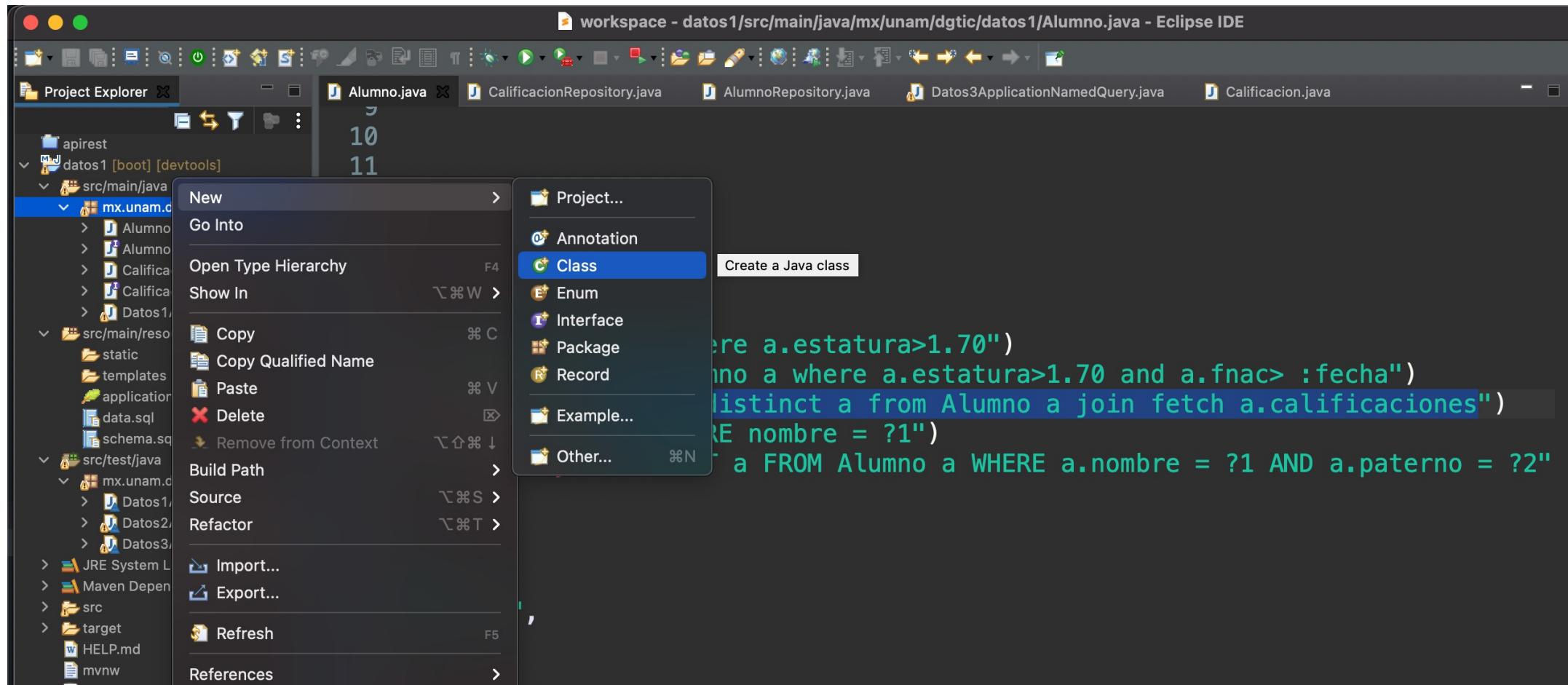


DTO Características

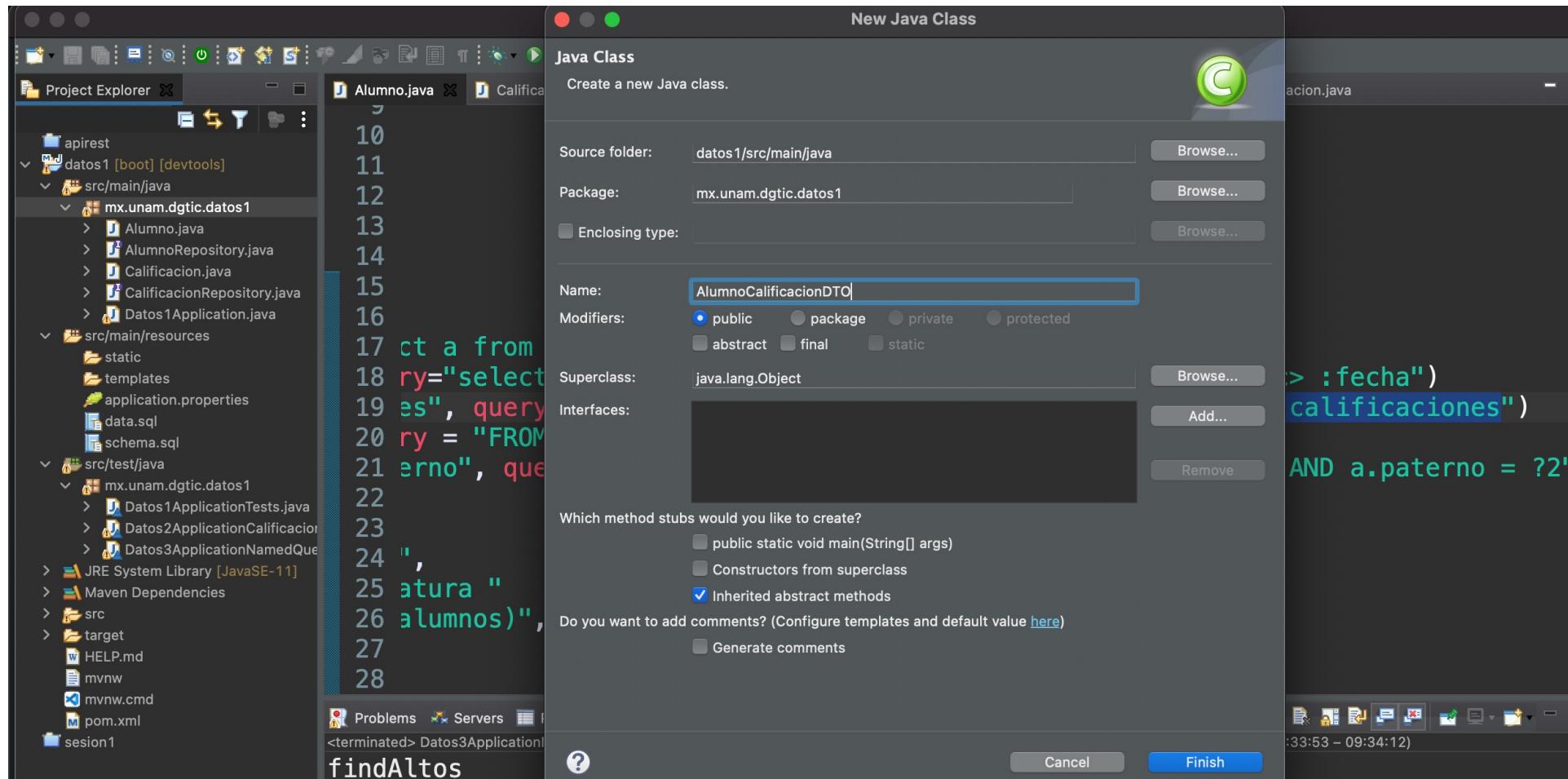
- **Solo lectura**
 - Dado que el objetivo de un DTO es utilizarlo como un objeto de transferencia entre el cliente y el servidor, es importante evitar tener operaciones de negocio o métodos que realicen cálculos sobre los datos, es por ello que solo se deben tener los métodos GET y SET de los respectivos atributos del DTO.
- **Serializable**
 - Si los objetos tendrán que viajar por la red, deberán de poder ser serializables, pero no solamente la clase en sí, sino que también todos los atributos que contenga el DTO deberán ser fácilmente serializables.



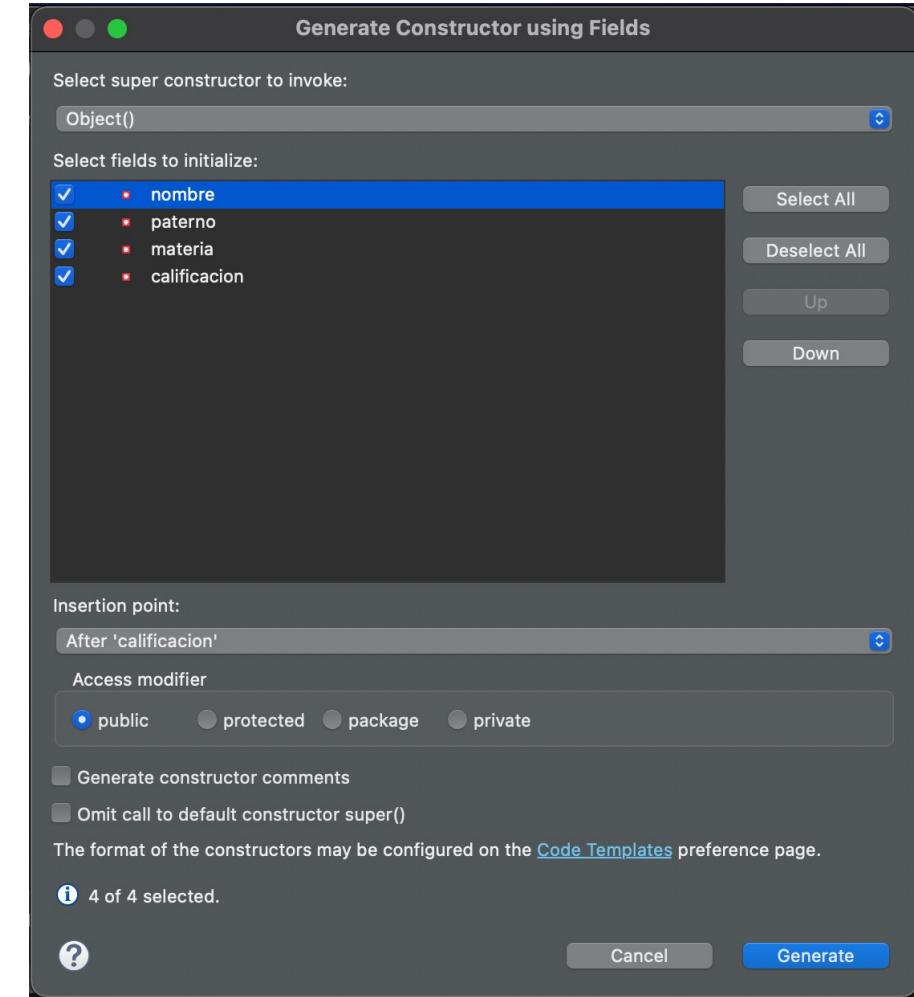
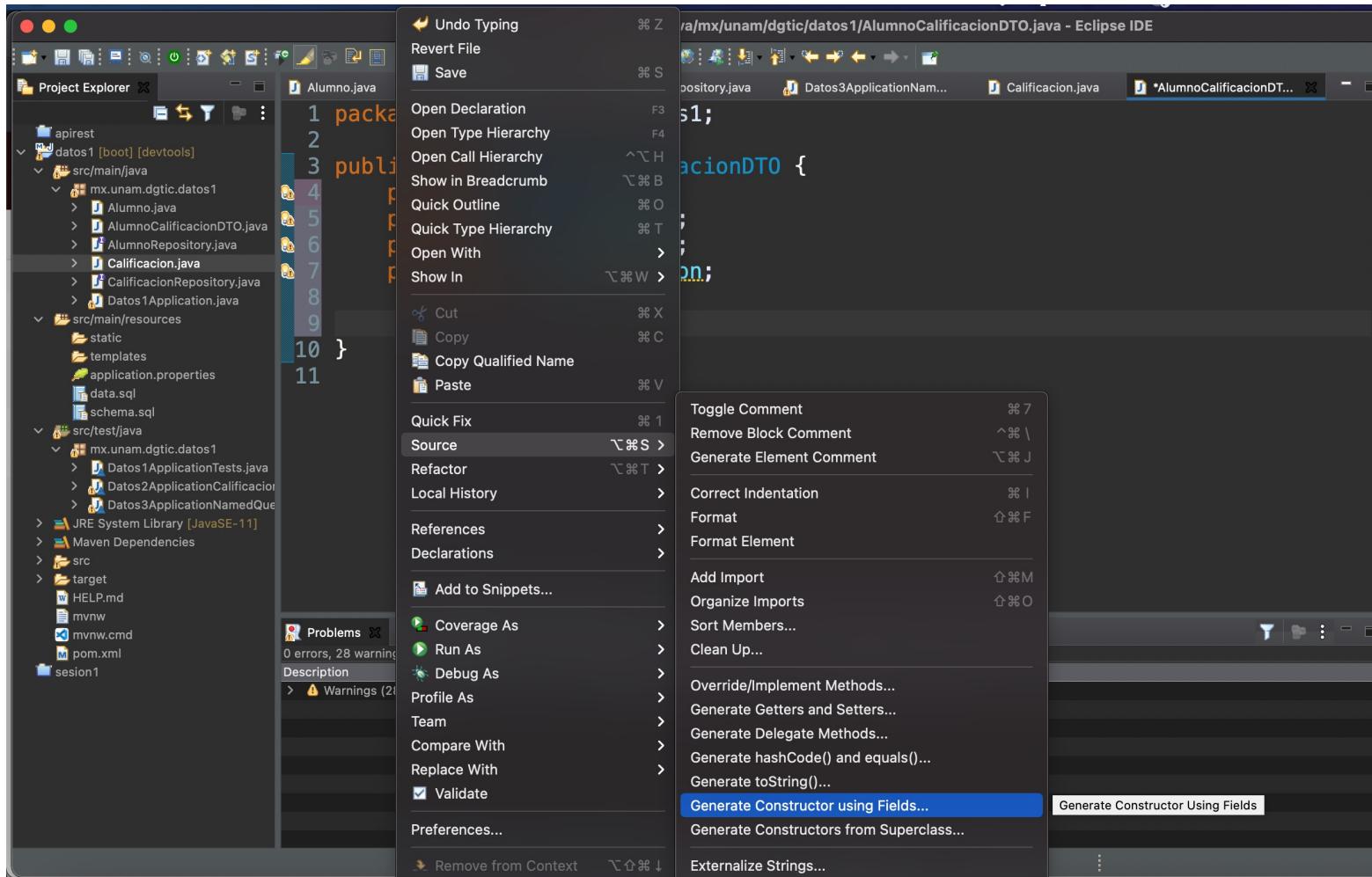
Crear la clase DTO



Crear la clase DTO



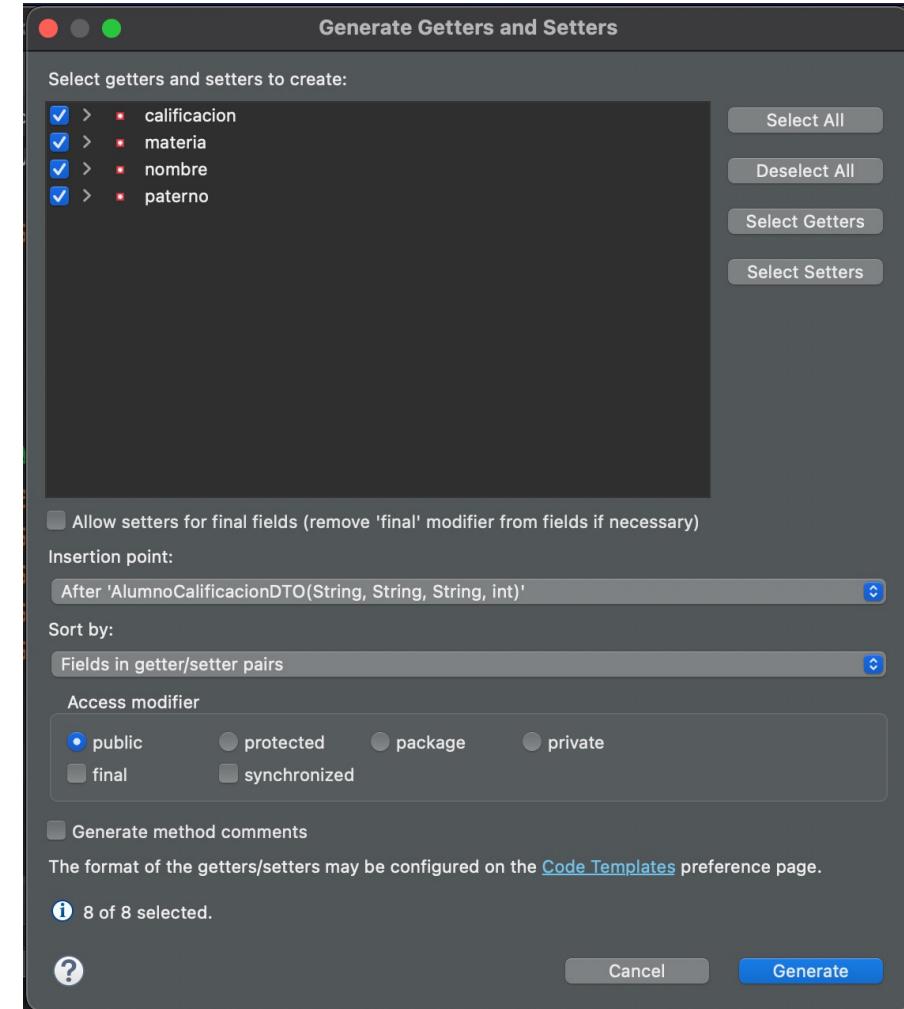
Crear la clase DTO



Crear la clase DTO

The screenshot shows the Eclipse IDE interface. On the left is the Project Explorer view, displaying a Java project structure. In the center is the code editor showing a Java class named `Alumno.java`. A context menu is open over the code, with the "Generate Getters and Setters..." option highlighted in blue. A tooltip for this option is visible, stating "Generate Getter and Setter Methods for the Type's Fields". The menu also includes other options like "Override/Implement Methods...", "Generate Delegate Methods...", and "Generate hashCode() and equals()...".

```
s1;
accionDTO {
;
;
on;
ionDTO(String nombre, String paterno, String materia,
re;
```



AlumnoRepository

AlumnoRepository

findAlumnoCalificacionDTO()



AlumnoRepository

```
//DTO
@Query (value = "select distinct "
        + "new mx.unam.dgtic.datos1.AlumnoCalificacionDTO("
        + "a.nombre, a.paterno, c.materia, c.calificacion) "
        + "from Alumno a, Calificacion c "
        + "where a.matricula = c.alumno"
        )
public List<AlumnoCalificacionDTO> findAlumnoCalificacionDTO();
```



Ordenamiento de resultados

- Puede aplicarse una ordenación estática usando la cláusula ***OrderBy*** después de la propiedad de búsqueda en el nombre del método de consulta que hace referencia a una propiedad y proporcionando una dirección de ordenación (Asc o Desc).
- `public List<Alumno> findByOrderByNombre();`
- `public List<Alumno> findByOrderByNombreAscPaternoDesc();`

AlumnoRepository

AlumnoRepository

findByOrderByNombre()

findByOrderByNombreAscPaternoDesc()

Paging And Sorting



Paginaciones y
ordenaciones

AlumnoRepository

AlumnoRepository

findAll(Sort ordenacion)



AlumnoRepository

```
public interface AlumnoRepository extends  
PagingAndSortingRepository<Alumno, String> {  
}
```



AlumnoRepository

The screenshot shows a Java code editor in an IDE. The code being written is:

```
61 }
62
63 @Test
64 void buscarTodosOrdenadosParametro() {
65
66     List<Alumno> lista = repositorioAlumno.findAll();
67     System.out.print
68     lista.forEach(Sy
69
70     lista = reposito
71     System.out.print
72     lista.forEach(Sy
73     assertNotNull(li
74
75
76     Iterable<Alumno> it= repositorioAlumno.findAll();
77     List<Alumno> lista2= new ArrayList<Alumno>();
78     it.forEach(lista2::add);
79     List<Alumno> listaOrdenada=lista2.stream()
```

A code completion tooltip is displayed over the call to `repositorioAlumno.findAll()`. The tooltip provides the following information:

- Returns all entities sorted by the given options.
- Parameters:
 - sort
- Returns:
 - all entities sorted by the given options

The tooltip also lists several other methods from the `AlumnoRepository` interface:

- `findAllAlumnosNative() : List<Alumno>` - `AlumnoRepository`
- `findAllWithCalificaciones() : List<Alumno>` - `AlumnoRepository`
- `findAll() : Iterable<Alumno>` - `CrudRepository`
- `findAll(Pageable pageable) : Page<Alumno>` - `PagingAndSortingRepository`
- `findAll(Sort sort) : Iterable<Alumno>` - `PagingAndSortingRepository`
- `findAllById(Iterable<String> ids) : Iterable<Alumno>` - `CrudRepository`

At the bottom of the tooltip, there are instructions: "Press 'Tab' from proposal table or click for focus" and "Press '^Space' to show Template Proposals".

Ordenamiento

Sort

By

nombre

Ascending



Ordenamiento anidado

Sort

By

nombre

Ascending

And

Sort

By

paterno

Descending



Ordenamiento anidado

```
@Test
void buscarTodosOrdenadosParametroAnidado() {

    Iterable<Alumno> it = repositorioAlumno.findAll(Sort.by("nombre")
        .ascending().and(Sort.by("paterno").ascending()));
    List<Alumno> milista = new ArrayList<Alumno>();

    System.out.println("findAll(Sort.by(\"nombre\")).descending().and(Sort.
        it.forEach(milista::add);
    it.forEach(System.out::println);

    List<Alumno> listaOrdenada=milista.stream()
        .sorted(Comparator.comparing(Alumno::getNombre)
        .thenComparing(Alumno::getPaterno))
        .collect(Collectors.toList());
    assertEquals(milista,listaOrdenada);
}
```



TypedSort

- Para una forma más segura de tipos para definir expresiones de ordenación, utilizar la entidad para la cual se define la expresión de ordenación y usar referencias de métodos para definir las propiedades en las que ordenar.

```
TypedSort<Alumno> ordenar= Sort.sort(Alumno.class);
Sort ordenFinal = ordenar.by(Alumno::getNombre).ascending()
.and(ordenar.by(Alumno::getPaterno)).ascending();
```



TypedSort

TypeSort

<Alumno>

alumnoOrden

Sort.sort

Alumno.class

Sort

By

Alumno::nombre

Ascending

And

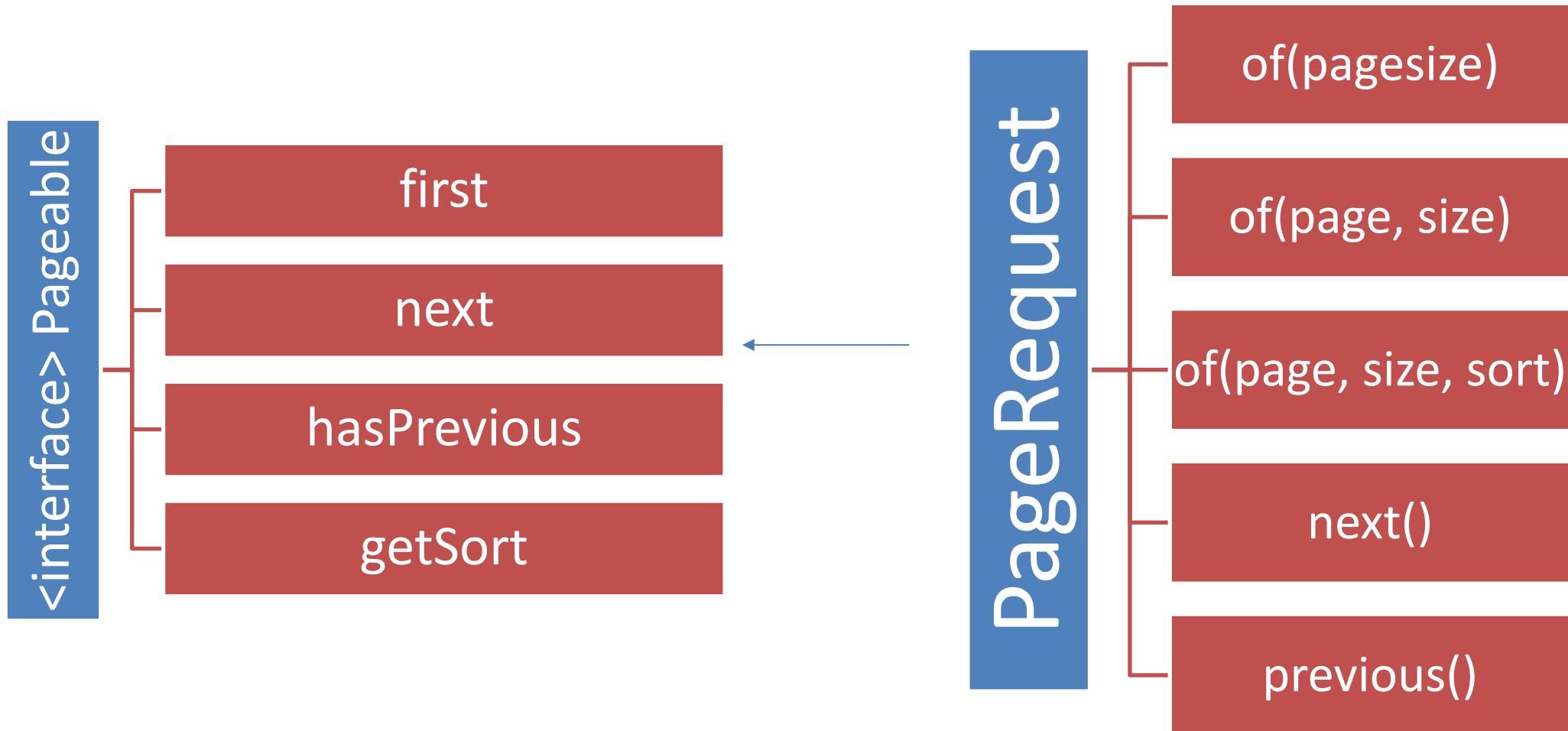
Sort

By

Alumno::paterno

Descending

Paginación



AlumnoRepository

AlumnoRepository

findAll(Pageable)

findAll(Sort)



Paginación

Page

Page

Page

anterior

siguiente



Contacto

Dr. Omar Mendoza González

omarmendoza564@aragon.unam.mx

