

Oswaldo Cabrera Pérez

Ejercicios 2.- Spring Data CrudRepository alumno y calificaciones

En nuestra carpeta Modelo creamos nuestras clases Alumno y Calificación. Nos apoyamos de Lombok para que los métodos que usamos prácticamente s

```
10
11 88 usages OswaCabrera
12 @NoArgsConstructor
13 @AllArgsConstructor
14 @Getter
15 @Setter
16 @Entity
17 @Table(name = "alumnos")
18 //Named Queries
19 @NamedQuery(name="Alumno.buscarAltos",
20             query="SELECT a FROM Alumno a WHERE a.estatura > 1.60")
21 public class Alumno {
22     @Id
23     private String matricula;
24     @Column(name = "nombre_alumno")
25     private String nombreAlumno;
26
27     private String paterno;
28     private Date fnac;
29     private double estatura;
30
31     @OneToMany(fetch=FetchType.EAGER, mappedBy = "alumno")
32     private List<Calificacion> calificaciones = new ArrayList<>();
33
34 OswaCabrera
35 @Override
36 public String toString() {
37     return "Alumno{" +
38         "matricula='" + matricula + '\'' +
39         ", nombreAlumno='" + nombreAlumno + '\'' +
40         ", paterno='" + paterno + '\'' +
41         ", fnac='" + fnac +
42         ", estatura=" + estatura +
43         '}'
44     }
45 }
```

```
.java Calificacion.java x data.sql schema_al_cal.sql appl
4 > import ...
6
22 usages 1 OswaCabrera *
7 @AllArgsConstructor
8 @NoArgsConstructor
9 @Setter
10 @Getter
11 @EqualsAndHashCode
12 @Entity
13 @Table(name = "calificaciones")
14 public class Calificacion {
15     @Id
16     private Integer id;
17     private String materia;
18     private int calificacion;
19     //private String alumnosMatricula;
20     @ManyToOne
21     @JoinColumn(name = "alumnos_matricula")
22     private Alumno alumno;
23
24     1 OswaCabrera *
25     @Override
26     public String toString() {
27         return "Materia Info: " +
28             "\n- ID: " + id +
29             "\n- Materia: " + materia +
30             "\n- Calificación: " + calificacion +
31             "\n- " + (alumno != null ? alumno.toString()
32                 | "\n" : "No asignado");
33     }
}
```

Para cada modelo creamos la interfaz Repository, AlumnoRepository y CalificacionRepository, ambas clases extienden a la interfaz CrudRepository la cual hace que nuestras operaciones de CRUD se implementen en automático.

Calificacion.java AlumnoRepository.java schema_al_cal.sql applicat

6 usages OswaCabrera

```

public interface AlumnoRepository extends CrudRepository<Alumno, String> {
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumno(String nom);
    1 usage OswaCabrera
    List<Alumno> findByPaterno(String pat);
    1 usage OswaCabrera
    List<Alumno> findByEstatura(double estatura);
    1 usage OswaCabrera
    List<Alumno> findByFnac(Date fnac);
    1 usage OswaCabrera
    List<Alumno> findByNombreAlumnoEquals(String nom);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoNot(String nom);
    1 usage OswaCabrera
    List<Alumno> findByPaternoIsNull();
    1 usage OswaCabrera
    List<Alumno> findByPaternoIsNotNull();
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoStartingWith(String prefijo);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoEndingWith(String posfijo);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoContaining(String infijo);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoLike(String like);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByEstaturaLessThan(double estatura);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByEstaturaLessThanEqual(double estatura);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByEstaturaBetween(Double init, Double stop);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByEstaturaIn(Collection<Double> alumnos);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByFnacAfter(Date fnac);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByFnacBefore(Date fnac);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoOrPaterno(String nombre, String paterno);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByNombreAlumnoAndPaterno(String nombre, String paterno);
    1 usage OswaCabrera
    List<Alumno> findAlumnoByEstaturaOrderByEstatura(Double estatura);
    1 usage OswaCabrera

```

```
1 usage  OswaCabrera
32      List<Alumno> findAlumnoByEstaturaOrderByFnac(Double estatura);
33      //Named Queries
1 usage  OswaCabrera
34      List<Alumno> buscarAltos();
```

```
Calificacion.java  AlumnoRepository.java  CalificacionRepository.java  application.pro
package mx.unam.dgtic.alumnocrud.repository;

import ...

2 usages  OswaCabrera
public interface CalificacionRepository extends CrudRepository<Calificacion, Integer>{
    1 usage  OswaCabrera
    List<Calificacion> findByMateria(String materia);
    1 usage  OswaCabrera
    List<Calificacion> findCalificacionByCalificacionEquals(Integer calif);
    1 usage  OswaCabrera
    List<Calificacion> findCalificacionByAlumnoMatricula(String matricula);
    1 usage  OswaCabrera
    List<Calificacion> findCalificacionByAlumnoPaterno(String paterno);
    1 usage  OswaCabrera
    List<Calificacion> findCalificacionByAlumnoNombreAlumno(String nombre);
    1 usage  OswaCabrera
    List<Calificacion> findCalificacionByAlumnoNombreAlumnoAndAlumnoPaterno(String n, String p);
}
```

Para comprobar el correcto funcionamiento de nuestro CRUD realizamos pruebas unitarias. Tanto para la clase alumno:

✓ AlumnosDerivadasTest (mx.unam.dgtic.al 398 ms)

✓ buscarPorPaternoTest() 317 ms

✓ buscarPorNombreLikeTest() 5 ms

✓ buscarLosNacidosDespuesTest() 6 ms

✓ buscarPorFnacTest() 4 ms

✓ buscarPorNombreContainingTest() 5 ms

✓ buscarLosNacidosAntesTest() 3 ms

✓ buscarPorEstaturaInTest() 5 ms

✓ buscarPorNombreOrPaternoTest() 4 ms

✓ buscarPorNombreEndingWithTest() 4 ms

✓ buscarPorEstaturaOrderByFnacTest() 3 ms

✓ buscarPorNombreStartingWithTest() 2 ms

✓ buscarPorNombreNotTest() 5 ms

✓ buscarPorPaternolsNullTest() 3 ms

✓ buscarPorEstaturaBetweenTest() 5 ms

✓ buscarPorEstaturaTest() 3 ms

✓ buscarAlumnosByNombre() 3 ms

✓ buscarPorEstaturaLessTest() 3 ms

✓ buscarPorNombreEqualsTest() 3 ms

✓ buscarPorEstaturaLessEqualTest() 5 ms

✓ buscarPorEstaturaOrderByEstaturaTest() 3 ms

✓ buscarPorPaternolsNotNullTest() 7 ms

✓ Tests passed: 21 of 21 tests - 398 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap classpath has been appended

Oswaldo Cabrera Pérez

Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Run AlumnosDerivadasTest x

✓ AlumnosDerivadasTest (mx.unam.dgtic.al 408 ms)

✓ buscarPorPaternoTest() 315 ms

✓ buscarPorNombreLikeTest() 5 ms

✓ buscarLosNacidosDespuesTest() 5 ms

✓ buscarPorFnacTest() 4 ms

✓ buscarPorNombreContainingTest() 5 ms

✓ buscarLosNacidosAntesTest() 9 ms

✓ buscarPorEstaturaInTest() 5 ms

✓ buscarPorNombreOrPaternoTest() 6 ms

✓ buscarPorNombreEndingWithTest() 4 ms

✓ buscarPorEstaturaOrderByFnacTest() 4 ms

✓ buscarPorNombreStartingWithTest() 2 ms

✓ buscarPorNombreNotTest() 4 ms

✓ buscarPorPaternolsNullTest() 3 ms

✓ buscarPorEstaturaBetweenTest() 5 ms

✓ buscarPorEstaturaTest() 3 ms

✓ buscarAlumnosByNombre() 3 ms

✓ buscarPorEstaturaLessTest() 3 ms

✓ buscarPorNombreEqualsTest() 3 ms

✓ buscarPorEstaturaLessEqualTest() 4 ms

✓ buscarPorEstaturaOrderByEstaturaTest() 11 ms

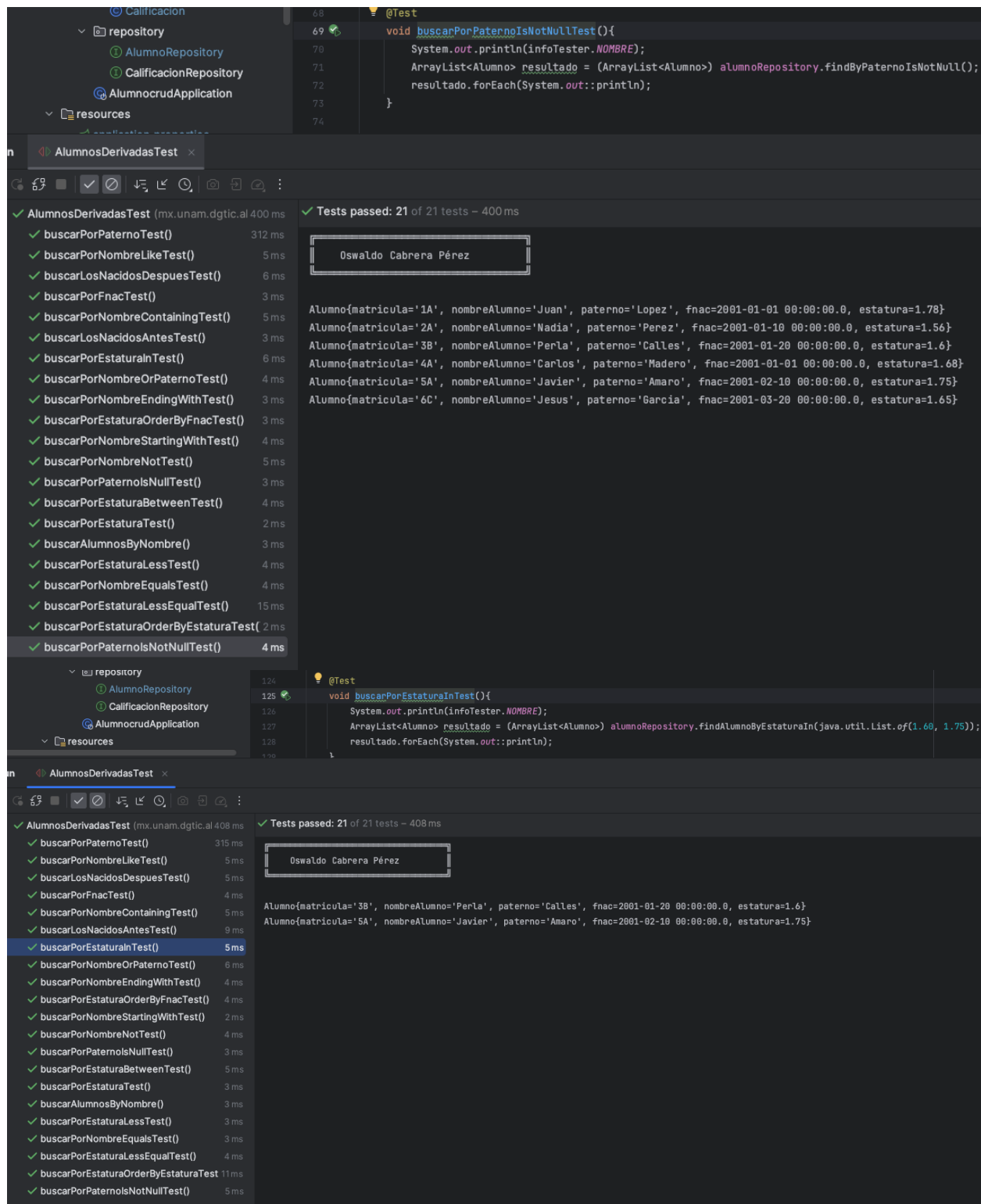
✓ buscarPorPaternolsNotNullTest() 5 ms

✓ Tests passed: 21 of 21 tests - 408 ms

Oswaldo Cabrera Pérez

Alumno{matricula='6C', nombreAlumno='Jesus', paterno='Garcia', fnac=2001-03-20 00:00:00.0, estatura=1.65}

Alumno{matricula='7B', nombreAlumno='Gema', paterno='null', fnac=2001-03-20 00:00:00.0, estatura=1.53}



repository

AlumnoRepository

CalificacionRepository

AlumnocrudApplication

resources

76

77

78

79

80

81

```

void buscarPorNombreStartingWithTest(){
    System.out.println(INFO + "TESTER: " + NOMBRE);
    ArrayList<Alumno> resultado = (ArrayList<Alumno>) alumnoRepository.findAlumnoByNombreAlumnoStartingWith( prefijo: "J");
    resultado.forEach(System.out::println);
}

```

Run

AlumnosDerivadasTest

AlumnosDerivadasTest (mx.unam.dgtic.al 400 ms)

✓ buscarPorPaternoTest()

312 ms

✓ buscarPorNombreLikeTest()

5 ms

✓ buscarLosNacidosDespuesTest()

6 ms

✓ buscarPorFnacTest()

3 ms

✓ buscarPorNombreContainingTest()

5 ms

✓ buscarLosNacidosAntesTest()

3 ms

✓ buscarPorEstaturaInTest()

6 ms

✓ buscarPorNombreOrPaternoTest()

4 ms

✓ buscarPorNombreEndingWithTest()

3 ms

✓ buscarPorEstaturaOrderByFnacTest()

3 ms

✓ buscarPorNombreStartingWithTest()

4 ms

✓ buscarPorNombreNotTest()

5 ms

✓ buscarPorPaternoIsNotNullTest()

3 ms

✓ buscarPorEstaturaBetweenTest()

4 ms

✓ buscarPorEstaturaTest()

2 ms

✓ buscarAlumnosByNombre()

3 ms

✓ buscarPorEstaturaLessTest()

4 ms

✓ buscarPorNombreEqualsTest()

4 ms

✓ buscarPorEstaturaLessEqualTest()

15 ms

✓ buscarPorEstaturaOrderByEstaturaTest()

2 ms

✓ buscarPorPaternoIsNotNullTest()

4 ms

Tests passed: 21 of 21 tests - 400 ms

Oswaldo Cabrera Pérez

Alumno{matricula='1A', nombreAlumno='Juan', paterno='Lopez', fnac=2001-01-01 00:00:00.0, estatura=1.78}

Alumno{matricula='5A', nombreAlumno='Javier', paterno='Amaro', fnac=2001-02-10 00:00:00.0, estatura=1.75}

Alumno{matricula='6C', nombreAlumno='Jesus', paterno='Garcia', fnac=2001-03-20 00:00:00.0, estatura=1.65}

Como para la clase Materia

ConsultaCalificacionesTest

AlumnoRepository

CalificacionRepository

AlumnocrudApplication

resources

76

77

78

79

80

81

```

void buscarPorNombreStartingWithTest(){
    System.out.println(INFO + "TESTER: " + NOMBRE);
    ArrayList<Alumno> resultado = (ArrayList<Alumno>) alumnoRepository.findAlumnoByNombreAlumnoStartingWith( prefijo: "J");
    resultado.forEach(System.out::println);
}

```

Run

ConsultaCalificacionesTest

ConsultaCalificacionesTest (mx.unam.dgtic.al 430 ms)

✓ buscarPorPaternoTest()

385 ms

✓ buscarCalificacionByNombreYPaterno()

9 ms

✓ buscarCalificacionByCalificacionEquals()

12 ms

✓ buscarPorNombreTest()

6 ms

✓ buscarCalificacionesByMateria()

10 ms

✓ buscarCalificacionByMatriculaEquals()

8 ms

Tests passed: 6 of 6 tests - 430 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap

Oswaldo Cabrera Pérez

Buscar por paterno

Materia Info:

- ID: 1

- Materia: BD

- Calificación: 10

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 2

- Materia: PD0

- Calificación: 9

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 3

- Materia: SI

- Calificación: 10

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 4

- Materia: BDII

- Calificación: 8

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

session3

AlumnosDerivadasTest

session4

CalificacionesAlumnoTest

ConsultaCalificacionesTest

session5

AlumnoNamedQueriesTest

44

45

46

47

48

49

50

OswaCabrera*

@Test

void buscarPorPaternoTest(){

System.out.println(infoTester.NOMBRE);

System.out.println("Buscar por paterno");

ArrayList<Calificacion> cals = (ArrayList<Calificacion>) calificacionRepository.findCalificacion

cals.forEach(System.out::println);

}

run

ConsultaCalificacionesTest

Tests passed: 6 of 6 tests - 335 ms

Java HotSpot(TM) 64-Bit Server VM warning: Sharing is only supported for boot loader classes because bootstrap

Oswaldo Cabrera Pérez

Buscar por paterno

Materia Info:

- ID: 1

- Materia: BD

- Calificación: 10

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 2

- Materia: P00

- Calificación: 9

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 3

- Materia: SI

- Calificación: 10

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 4

java

mx.unam.dgtic.alumnocrud

sesion3

AlumnosDerivadasTest

sesion4

CalificacionesAlumnoTest

ConsultaCalificacionesTest

session5

AlumnoNamedQueriesTest

26

27

28

29

30

31

32

33

34

35

@Test

void buscarCalificacionByCalificacionEquals(){

System.out.println(infoTester.NOMBRE);

System.out.println("Buscar calificaciones igual a 8");

ArrayList<Calificacion> resultado =

(ArrayList<Calificacion>) calificacionRepository.findCalificacionByCalificacionEquals(8);

resultado.forEach(System.out::println);

}

OswaCabrera

@Test

ConsultaCalificacionesTest

Tests passed: 6 of 6 tests - 335 ms

Oswaldo Cabrera Pérez

Buscar calificaciones igual a 8

Materia Info:

- ID: 4

- Materia: BDII

- Calificación: 8

- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 7

- Materia: BDII

- Calificación: 8

- Alumno{matricula='7B', nombreAlumno='Gema', paterno='null', fnac=2001-03-20 00:00:00.0, estatura=1.53}

test

- java
 - mx.unam.dgtic.alumnocrud
 - sesion3
 - AlumnosDerivadasTest
 - sesion4

```
18 // @Test
19 void buscarCalificacionesByMateria(){
20     System.out.println(infoTester.NOMBRE);
21     System.out.println("Buscar por P00");
22     ArrayList<Calificacion> resultado = (ArrayList<Calificacion>) calificacionRepository.findByMateria("P00");
23     resultado.forEach(System.out::println);
24 }
```

Run ConsultaCalificacionesTest x

ConsultaCalificacionesTest (mx.unam.dgtic) 335 ms

- ✓ buscarPorPaternoTest() 316 ms
- ✓ buscarCalificacionByNombreYPaterno() 4 ms
- ✓ buscarCalificacionByCalificacionEquals() 4 ms
- ✓ buscarPorNombreTest() 2 ms
- ✓ buscarCalificacionesByMateria() 6 ms
- ✓ buscarCalificacionByMatriculaEquals() 3 ms

Tests passed: 6 of 6 tests - 335 ms

Oswaldo Cabrera Pérez

Buscar por P00

Materia Info:

- ID: 2
- Materia: P00
- Calificación: 9
- Alumno{matricula='2A', nombreAlumno='Nadia', paterno='Perez', fnac=2001-01-10 00:00:00.0, estatura=1.56}

Materia Info:

- ID: 6
- Materia: P00
- Calificación: 5
- Alumno{matricula='3B', nombreAlumno='Perla', paterno='Calles', fnac=2001-01-20 00:00:00.0, estatura=1.6}

Materia Info:

- ID: 8
- Materia: P00
- Calificación: 5
- Alumno{matricula='7B', nombreAlumno='Gema', paterno='null', fnac=2001-03-20 00:00:00.0, estatura=1.53}

Materia Info:

- ID: 10
- Materia: P00
- Calificación: 5
- Alumno{matricula='6C', nombreAlumno='Jesus', paterno='Garcia', fnac=2001-03-20 00:00:00.0, estatura=1.65}

screen_51_car.sql

test

- java
 - mx.unam.dgtic.alumnocrud
 - sesion3
 - AlumnosDerivadasTest
 - sesion4

```
19 // @Test
20 void buscarCalificacionByNombreYPaterno(){
21     System.out.println(infoTester.NOMBRE);
22     System.out.println("Buscar por nombre y paterno");
23     ArrayList<Calificacion> cals = (ArrayList<Calificacion>) calificacionRepository.findCalificacionByAlumnoNombreAlumnoAndAlumnoPaterno("Perla", "Calles");
24     cals.forEach(System.out::println);
25 }
26 }
```

ConsultaCalificacionesTest x

ConsultaCalificacionesTest (mx.unam.dgtic) 358 ms

- ✓ buscarPorPaternoTest() 332 ms
- ✓ buscarCalificacionByNombreYPaterno() 6 ms
- ✓ buscarCalificacionByCalificacionEquals() 6 ms
- ✓ buscarPorNombreTest() 4 ms
- ✓ buscarCalificacionesByMateria() 6 ms
- ✓ buscarCalificacionByMatriculaEquals() 4 ms

Tests passed: 6 of 6 tests - 358 ms

Oswaldo Cabrera Pérez

Buscar por nombre y paterno

Materia Info:

- ID: 5
- Materia: B0
- Calificación: 9
- Alumno{matricula='3B', nombreAlumno='Perla', paterno='Calles', fnac=2001-01-20 00:00:00.0, estatura=1.6}

Materia Info:

- ID: 6
- Materia: P00
- Calificación: 5
- Alumno{matricula='3B', nombreAlumno='Perla', paterno='Calles', fnac=2001-01-20 00:00:00.0, estatura=1.6}