



LABORATORIO N° 09

**Desarrollo de
aplicaciones Web:
Aplicaciones web con
Spring Boot y MongoDB.**



DOCENTE:
Coello Palomino, Ricardo

CURSO:
Desarrollo de aplicaciones Web
5 - C24 - Sección A B C D E

DESARROLLO DE APLICACIONES WEB AVANZADO: Aplicaciones web con Spring Starters y Actuators

I. Capacidades

- Implementar una aplicación JavaSpring Boot 2.1 Actuators y Starters

II. Seguridad

- En este laboratorio está prohibida la manipulación del hardware, conexiones eléctricas o de red. Así como la ingesta de alimentos y bebidas.
- Ubicar maletines y/o mochilas en lugar destinado para tal fin.
- Dejar la mesa de trabajo y la silla utilizada limpias y ordenadas.

III. Fundamento teórico

- Revise el material de la semana correspondiente antes del desarrollo del laboratorio.

IV. Normas empleadas

- *No aplica*

V. Recursos

- En este laboratorio, cada estudiante trabajará con una computadora con Windows 10.
- La instalación del software requerido se realizará en el equipo virtual.

VI. Metodología para el desarrollo de la tarea

- El desarrollo del laboratorio es individual

VII. Procedimiento

Base de Datos

1.- Abrir el programa MongoDB

2.- Crear la Base de Datos Escuela

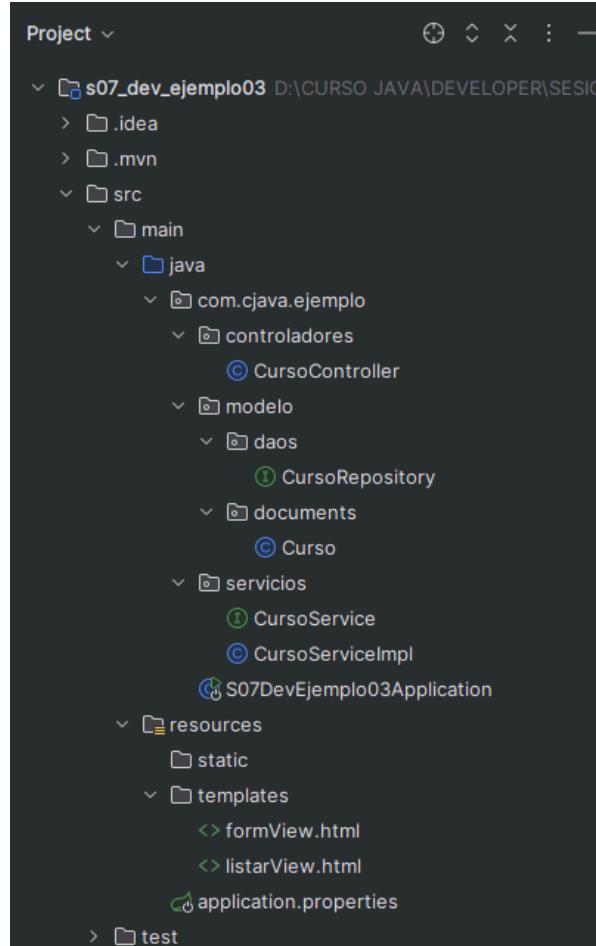
The screenshot shows the MongoDB Compass interface. On the left, the 'Connections' sidebar lists two connections: 'cone' and 'localhost:27017'. Under 'localhost:27017', there are several databases: 'Negocios2024', 'admin', 'comercial', 'config', 'data', 'datos', 'demo0IBD', 'escuela', and 'local'. The 'escuela' database is currently selected. Inside 'escuela', the 'Collection Name' dropdown is set to 'usuario'. The 'usuario' collection details are displayed: Storage size: 4.10 kB, Documents: 0, Avg. document size: 0 B, Indexes: 1, and Total index size: 4.10 kB. At the top right, there are buttons for 'Open MongoDB shell', 'Create collection', 'Refresh', and 'View'.

PROYECTO SPRING BOOT

3.- Crear el siguiente proyecto:

The screenshot shows the 'New Project' dialog from the Start Spring UI. In the left sidebar, under 'Generators', 'Spring Boot' is selected. The main form fields are: Server URL: 'start.spring.io', Name: 'PRJ_CRUD_SPRING_BOOT_MONGODB', Location: 'IVOS TECSUP/TECSUP/V_CICLO/LENGUAJE_PROGRAMACION_WEB/LABORATORIO_0', Language: 'Java', Type: 'Gradle - Groovy', Group: 'com.tecsup', Artifact: 'demo', Package name: 'com.tecsup.demo', JDK: '22 Oracle OpenJDK 22.0.2', Java: '22', and Packaging: 'Jar'. Below this, another 'New Project' dialog is partially visible, showing 'Spring Boot: 3.3.3' and a 'Dependencies' section with 'Developer Tools' and 'Web' listed.

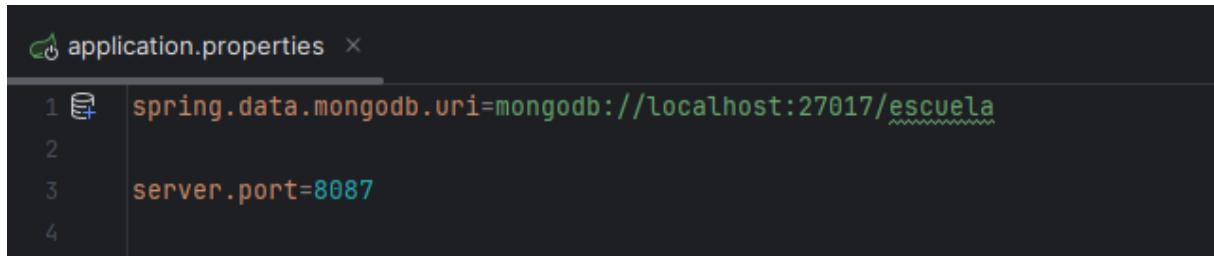
4.- Crear un Estructura de carpeta.



5.- Revisar el Archivo pom.xml

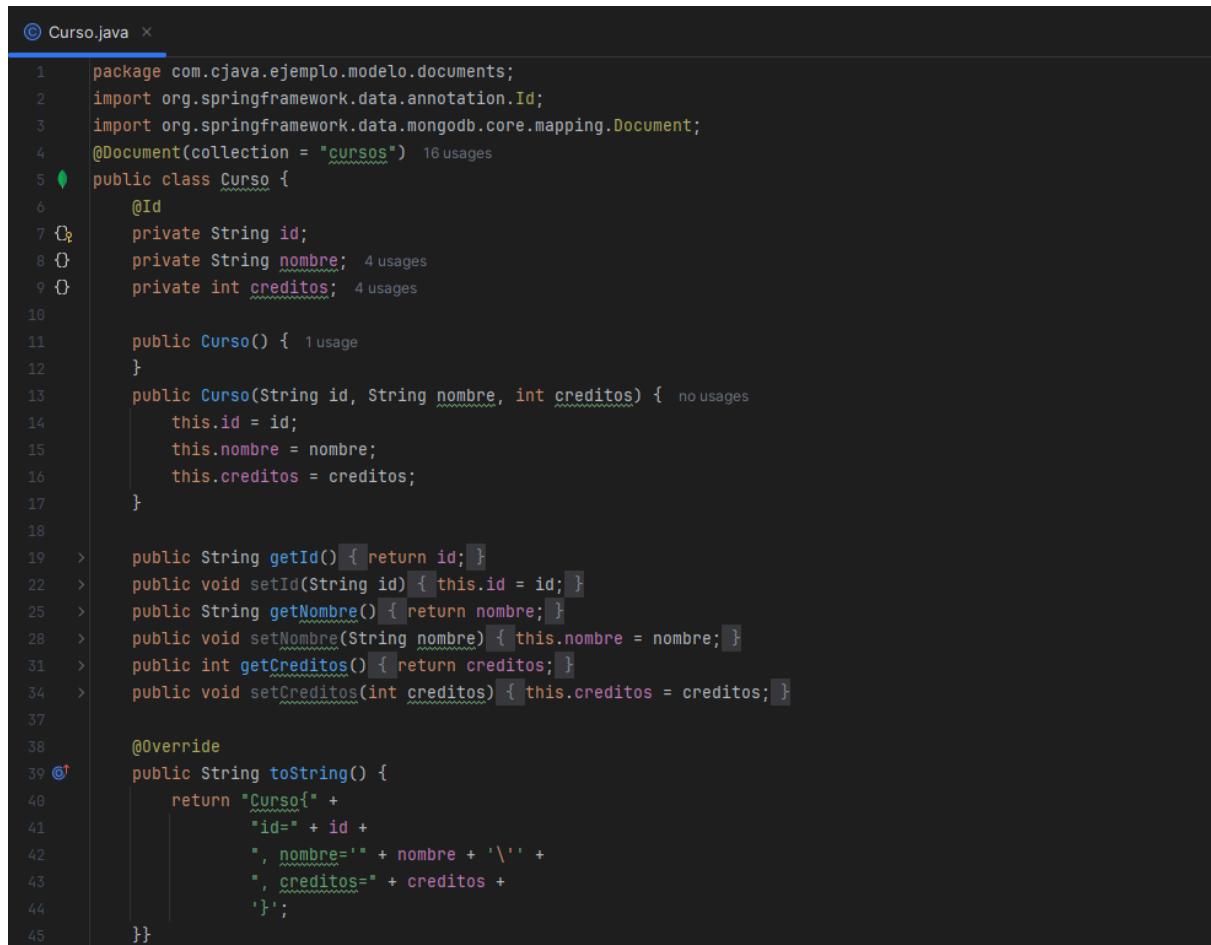
```
m pom.xml (s07_dev_ejemplo03) × ▲ 13
1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
3   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
4     <modelVersion>4.0.0</modelVersion>
5     <parent>
6       <groupId>org.springframework.boot</groupId>
7       <artifactId>spring-boot-starter-parent</artifactId>
8       <version>3.0.1</version>
9       <relativePath/> <!-- lookup parent from repository -->
10    </parent>
11    <groupId>com.cjava</groupId>
12    <artifactId>s07_dev_ejemplo03</artifactId>
13    <version>0.0.1-SNAPSHOT</version>
14    <name>s07_dev_ejemplo03</name>
15    <description>s07_dev_ejemplo03</description>
16    <properties>
17      <java.version>17</java.version>
18    </properties>
19    <dependencies> ⏺ Edit Starters...
20      <dependency>
21        <groupId>org.springframework.boot</groupId>
22        <artifactId>spring-boot-starter-data-mongodb</artifactId>
23      </dependency>
24      <dependency>
25        <groupId>org.springframework.boot</groupId>
26        <artifactId>spring-boot-starter-thymeleaf</artifactId>
27      </dependency>
28      <dependency>
29        <groupId>org.springframework.boot</groupId>
30        <artifactId>spring-boot-starter-web</artifactId>
31      </dependency>
32      <dependency>
33        <groupId>javax.validation</groupId>
34        <artifactId>validation-api</artifactId>
35        <version>2.0.1.Final</version>
36      </dependency>
37      <dependency>
38        <groupId>org.springframework.boot</groupId>
39        <artifactId>spring-boot-starter-test</artifactId>
40        <scope>test</scope>
41      </dependency>
42
43      <dependency>
44        <groupId>org.aspectj</groupId>
45        <artifactId>aspectjrt</artifactId>
46      </dependency>
47    </dependencies>
48
49    <build>
50      <plugins>
51        <plugin>
52          <groupId>org.springframework.boot</groupId>
53          <artifactId>spring-boot-maven-plugin</artifactId>
54        </plugin>
55
56        <plugin>
57          <groupId>org.apache.maven.plugins</groupId>
58          <artifactId>maven-compiler-plugin</artifactId>
59          <configuration>
60            <source>17</source>
61            <target>17</target>
62          </configuration>
63        </plugin>
64      </plugins>
65    </build>
66
67  </project>
```

6.- Configurar en la Application.properties



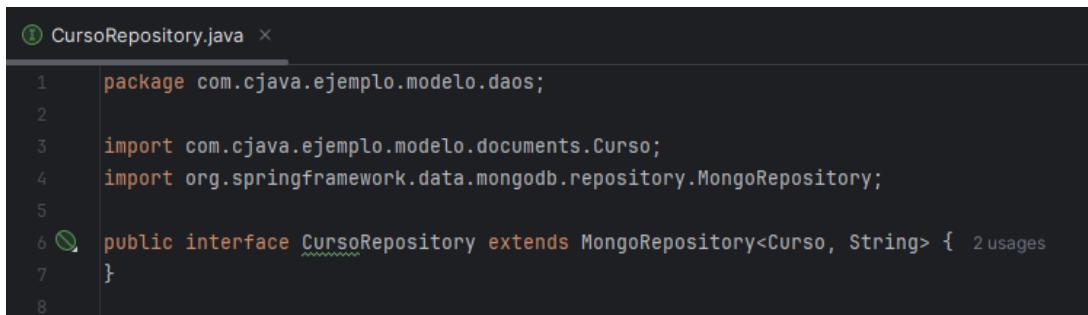
```
application.properties
1 spring.data.mongodb.uri=mongodb://localhost:27017/escuela
2
3 server.port=8087
4
```

7.- Clase Curso



```
Curso.java
1 package com.cjava.ejemplo.modelo.documents;
2 import org.springframework.data.annotation.Id;
3 import org.springframework.data.mongodb.core.mapping.Document;
4 @Document(collection = "cursos") 16 usages
5 public class Curso {
6     @Id
7     private String id;
8     private String nombre; 4 usages
9     private int creditos; 4 usages
10
11     public Curso() { 1 usage
12     }
13     public Curso(String id, String nombre, int creditos) { no usages
14         this.id = id;
15         this.nombre = nombre;
16         this.creditos = creditos;
17     }
18
19     >     public String getId() { return id; }
20     >     public void setId(String id) { this.id = id; }
21     >     public String getNombre() { return nombre; }
22     >     public void setNombre(String nombre) { this.nombre = nombre; }
23     >     public int getCreditos() { return creditos; }
24     >     public void setCreditos(int creditos) { this.creditos = creditos; }
25
26     @Override
27     public String toString() {
28         return "Curso{" +
29             "id=" + id +
30             ", nombre='" + nombre + '\'' +
31             ", creditos=" + creditos +
32             '}';
33     }
34 }
```

8.- Interfaz CursoRepository



```
CursoRepository.java
1 package com.cjava.ejemplo.modelo.daos;
2
3 import com.cjava.ejemplo.modelo.documents.Curso;
4 import org.springframework.data.mongodb.repository.MongoRepository;
5
6 public interface CursoRepository extends MongoRepository<Curso, String> { 2 usages
7 }
```

9.- Clase CursoService

```
① CursoService.java ×
1 package com.cjava.ejemplo.servicios;
2
3 import com.cjava.ejemplo.modelo.documents.Curso;
4
5 import java.util.List;
6
7 ② public interface CursoService { 3 usages 1 implementation
8
9     ③     public List<Curso> listar(); 2 usages 1 implementation
10
11    ④     public void grabar(Curso curso); 1 usage 1 implementation
12
13    ⑤     public Curso buscar(String id); 2 usages 1 implementation
14
15    ⑥     public void eliminar(String id); 1 usage 1 implementation
16
17 }
```

10.- Clase CursoServiceImpl

```
© CursoServiceImpl.java ×
1 package com.cjava.ejemplo.servicios;
2
3 import com.cjava.ejemplo.modelo.documents.Curso;
4 import com.cjava.ejemplo.modelo.daos.CursoRepository;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Service;
7 import org.springframework.transaction.annotation.Transactional;
8
9 import java.util.List;
10
11 @Service
12 public class CursoServiceImpl implements CursoService {
13
14     @Autowired
15     ③     private CursoRepository dao;
16
17     @Override 1 usage
18     ④     public void grabar(Curso curso) { dao.save(curso); }
19
20     @Override 1 usage
21     ⑤     public void eliminar(String id) { dao.deleteById(id); }
22
23     @Override 2 usages
24     @Transactional(readOnly = true)
25     ⑥     public Curso buscar(String id) { return dao.findById(id).orElse( other: null); }
26
27     @Override 2 usages
28     @Transactional(readOnly = true)
29     ⑦     public List<Curso> listar() { return (List<Curso>)dao.findAll(); }
30
31 }
```

11.- Clase CursoController

```
CursoController.java ×

1 package com.cjava.ejemplo.controladores;
2
3 import com.cjava.ejemplo.modelo.entidades.Curso;
4 import com.cjava.ejemplo.servicios.CursoService;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.stereotype.Controller;
7 import org.springframework.ui.Model;
8 import org.springframework.validation.BindingResult;
9 import org.springframework.web.bind.annotation.PathVariable;
10 import org.springframework.web.bind.annotation.RequestMapping;
11 import org.springframework.web.bind.annotation.RequestMethod;
12 import org.springframework.web.bind.annotation.SessionAttributes;
13 import org.springframework.web.bind.support.SessionStatus;
14
15 import javax.validation.Valid;
16 import java.util.Map;
17
18 @Controller
19 @SessionAttributes("curso")
20 public class CursoController {
21     @Autowired
22     private CursoService servicio;
23
24     @RequestMapping(value = @"/listar", method = RequestMethod.GET)
25     public String listar(Model model) {
26         model.addAttribute( attributeName: "titulo", attributeValue: "Listado de Cursos CJAVA");
27         model.addAttribute( attributeName: "cursos", servicio.listar());
28         return "listarView";
29     }
30
31     @RequestMapping(value = @"/form")
32     public String crear(Map<String, Object> model) {
33         Curso curso = new Curso();
34         model.put("curso", curso);
35         model.put("titulo", "Formulario de Curso");
36         return "formView";
37     }
38
39     @RequestMapping(value = @"/form/{id}")
40     public String editar(@PathVariable(value = "id") Integer id, Map<String, Object> model) {
41
42         Curso curso = null;
43
44         if (id > 0) {
45             curso = servicio.buscar(id);
46         } else {
47             return "redirect:/listar";
48         }
49         model.put("curso", curso);
50         model.put("titulo", "Editar Curso");
51         return "formView";
52     }
53
54     @RequestMapping(value = @"/form", method = RequestMethod.POST)
55     public String guardar(@Valid Curso curso, BindingResult result, Model model, SessionStatus status) {
56         if (result.hasErrors()) {
57             model.addAttribute( attributeName: "titulo", attributeValue: "Formulario de Curso");
58             return "formView";
59         }
60         //System.out.println(curso);
61         servicio.grabar(curso);
62         status.setComplete();
63         return "redirect:listar";
64     }
65
66     @RequestMapping(value = @"/eliminar/{id}")
67     public String eliminar(@PathVariable(value = "id") Integer id) {
68
69         if (id > 0) {
70             servicio.eliminar(id);
71         }
72         return "redirect:/listar";
73     }
74 }
```

12.- ListView

```
<> listView.html <
```

```
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3
4  <head>
5      <title>Title</title>
6      <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
7  </head>
8  <body>
9
10 <header>
11     <nav class="navbar navbar-expand-lg fixed-top" style="background-color: #1e7e34;">
12         <div class="container">
13             <a class="navbar-brand fs-3" href="#" style="color: white;">Spring Boot</a>
14             <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarScroll"
15                 aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">
16                 <span class="navbar-toggler-icon"></span>
17             </button>
18             <div class="collapse navbar-collapse" id="navbarScroll">
19                 <ul class="navbar-nav mx-auto my-2 my-lg-0 navbar-nav-scroll" style="--bs-scroll-height: 100px;">
20                     <li class="nav-item">
21                         <a class="nav-link link-light" aria-current="page" th:href="#">Inicio</a>
22                     </li>
23
24                     <li class="nav-item dropdown">
25                         <a class="nav-link link-light dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"
26                             aria-expanded="false">
27                             Gestión Cursos
28                         </a>
29                         <ul class="dropdown-menu">
30                             <li><a class="dropdown-item" th:href="@{/listar}">Cursos</a></li>
31                             <li><a class="dropdown-item" href="#">Credito</a></li>
32                         </ul>
33                     </li>
34
35                     <li class="nav-item dropdown">
36                         <a class="nav-link dropdown-toggle link-light" href="#" role="button" data-bs-toggle="dropdown"
37                             aria-expanded="false">
38                             Gestión Docente
39                         </a>
40                         <ul class="dropdown-menu">
41                             <li><a class="dropdown-item" th:href="@{/docente}">Docentes</a></li>
42                             <li><a class="dropdown-item" href="#">Especialidad</a></li>
43                         </ul>
44                     </li>
45
46                     <li class="nav-item dropdown">
47                         <a class="nav-link dropdown-toggle link-light" href="#" role="button" data-bs-toggle="dropdown"
48                             aria-expanded="false">
49                             Gestión Alumnos
50                         </a>
51                         <ul class="dropdown-menu">
52                             <li><a class="dropdown-item" href="#">Alumnos</a></li>
53                             <li><a class="dropdown-item" href="#">Apoderado</a></li>
54                         </ul>
55                     </li>
56                 </ul>
57
58                 <form class="d-flex" role="search">
59                     <input class="form-control me-2" type="search" placeholder="Buscar" aria-label="Search">
60                     <button class="btn btn-outline-light" type="submit">Búsqueda</button>
61                 </form>
62             </div>
63         </div>
64     </nav>
65 </header>
66
67 <div class="container" style="margin-top:90px ;" >
68     <h1>Listado de Cursos</h1>
69
70     <div style="padding: 10px;">
71         <p><a th:href="@{/form}" class="btn btn-success btn-xs">crear curso</a></p>
72     </div>

```

```

73
74     <table class="table table-primary table-hover table-bordered">
75         <thead class="table-light">
76             <tr style="text-align: center;">
77                 <th>CODIGO</th>
78                 <th>NOMBRES</th>
79                 <th>CREDITOS</th>
80                 <th>EDITAR</th>
81                 <th>ELIMINAR</th>
82             </tr>
83         </thead>
84         <tbody style="text-align: center;">
85             <tr th:each="curso: ${cursos}">
86                 <td th:text="${curso.id}"></td>
87                 <td th:text="${curso.nombre}"></td>
88                 <td th:text="${curso.creditos}"></td>
89
90                 <td><a class="btn btn-primary btn-xs" th:href="@{/form} + ${curso.id}" th:text="editar"></a></td>
91                 <td><a class="btn btn-danger btn-xs" th:href="@{/eliminar} + ${curso.id}" th:text="eliminar"
92                     onclick="return confirm('Estás seguro que quieres eliminar?');"></a></td>
93             </tr>
94         </tbody>
95     </table>
96 </div>
97
98 <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" ></script>
99
100 </body>
101 </html>

```

13.- formView

```

<> formView.html <
1  <!DOCTYPE html>
2  <html xmlns:th="http://www.thymeleaf.org">
3
4      <head>
5          <title>Title</title>
6          <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
7      </head>
8      <body>
9
10     <header>
11         <nav class="navbar navbar-expand-lg fixed-top" style="background-color: #1e7e34;">
12             <div class="container">
13                 <a class="navbar-brand fs-3" href="#" style="color: white;">Spring Boot</a>
14                 <button class="navbar-toggler" type="button" data-bs-toggle="collapse" data-bs-target="#navbarScroll"
15                     aria-controls="navbarScroll" aria-expanded="false" aria-label="Toggle navigation">
16                     <span class="navbar-toggler-icon"></span>
17                 </button>
18                 <div class="collapse navbar-collapse" id="navbarScroll">
19                     <ul class="navbar-nav mx-auto my-2 my-lg-0 navbar-nav-scroll " style="--bs-scroll-height: 100px;">
20                         <li class="nav-item">
21                             <a class="nav-link link-light" aria-current="page" th:href="@{[]}">Inicio</a>
22                         </li>
23
24                         <li class="nav-item dropdown">
25                             <a class="nav-link link-light dropdown-toggle" href="#" role="button" data-bs-toggle="dropdown"
26                               aria-expanded="false">
27                               Gestión Cursos
28                             </a>
29                             <ul class="dropdown-menu ">
30                                 <li><a class="dropdown-item" th:href="@{/listar}">Cursos</a></li>
31                                 <li><a class="dropdown-item" href="#">Credito</a></li>
32                             </ul>
33                         </li>

```

```
34
35         <li class="nav-item dropdown">
36             <a class="nav-link dropdown-toggle link-light" href="#" role="button" data-bs-toggle="dropdown"
37                 aria-expanded="false">
38                 Gestión Docente
39             </a>
40             <ul class="dropdown-menu">
41                 <li><a class="dropdown-item" th:href="@{/docente}">Docentes</a></li>
42                 <li><a class="dropdown-item" href="#">Especialidad</a></li>
43             </ul>
44         </li>
45
46         <li class="nav-item dropdown">
47             <a class="nav-link dropdown-toggle link-light" href="#" role="button" data-bs-toggle="dropdown"
48                 aria-expanded="false">
49                 Gestión Alumnos
50             </a>
51             <ul class="dropdown-menu">
52                 <li><a class="dropdown-item" href="#">Alumnos</a></li>
53                 <li><a class="dropdown-item" href="#">Apoderado</a></li>
54             </ul>
55         </li>
56     </ul>
57
58     <form class="d-flex" role="search">
59         <input class="form-control me-2" type="search" placeholder="Buscar" aria-label="Search">
60         <button class="btn btn-outline-light" type="submit">Búsqueda</button>
61     </form>
62     </div>
63 </div>
64 </nav>
65 </header>
66 <div class="container" style="margin-top:90px ;">
67     <h1>
68         Cursos
69     </h1>
70     <div th:object="${curso}" th:remove="tag">
71         <ul th:if="${#fields.hasErrors('*')}" class="alert alert-danger">
72             <li th:each="err: ${#fields.errors('*')} " th:text="${err}"></li>
73         </ul>
74     </div>
75     <form th:action="@{/form}" th:object="${curso}" method="post">
76         <div class="form-group row">
77             <label class="col-sm-2 col-form-label">Nombre</label>
78             <div class="col-sm-6">
79                 <input type="text" th:field="*{nombre}" class="form-control"
80                     th:errorclass="form-control alert-danger" /> <small
81                     class="form-text text-danger"
82                     th:if="${#fields.hasErrors('nombre')}" th:errors="*{nombre}"></small>
83             </div>
84         </div>
85         <div class="form-group row">
86             <label class="col-sm-2 col-form-label">Creditos</label>
87             <div class="col-sm-6">
88                 <input type="text" th:field="*{creditos}" class="form-control"
89                     th:errorclass="form-control alert-danger" /> <small
90                     class="form-text text-danger"
91                     th:if="${#fields.hasErrors('creditos')}" th:errors="*{creditos}"></small>
92             </div>
93         </div>
94
95         <div class="form-group row">
96             <div class="col-sm-6">
97                 <input type="submit" value="Crear Curso" class="btn btn-primary" />
98             </div>
99         </div>
100
101     </form>
102 </div>
103
104     <script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js" ></script>
105
106 </body>
107 </html>
```

14.- Ejecutar la aplicación

```

Run S07DevEjemplo03Application ×
Console Actuator
=====
:: Spring Boot :: (v3.0.1)

2024-09-14T21:51:14.510-05:00 INFO 17544 --- [main] c.c.ejemplo.S07DevEjemplo03Application : Starting S07DevEjemplo03Application using Java 17.0.12 with PID 17544 (D:\Users\user\IdeaProjects\S07DevEjemplo03\target\classes)
2024-09-14T21:51:14.513-05:00 INFO 17544 --- [main] c.c.ejemplo.S07DevEjemplo03Application : No active profile set, falling back to 1 default profile: "default"
2024-09-14T21:51:15.493-05:00 INFO 17544 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data MongoDB repositories in DEFAULT mode.
2024-09-14T21:51:15.551-05:00 INFO 17544 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning in 52 ms. Found 1 MongoDB repository.
2024-09-14T21:51:16.423-05:00 INFO 17544 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8087 (http)
2024-09-14T21:51:16.523-05:00 INFO 17544 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-09-14T21:51:16.524-05:00 INFO 17544 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10.1.4]
2024-09-14T21:51:16.842-05:00 INFO 17544 --- [main] o.a.c.c.C.[Tomcat].[localhost].[] : Initializing Spring embedded WebApplicationContext
2024-09-14T21:51:16.847-05:00 INFO 17544 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization completed in 2283 ms
2024-09-14T21:51:17.053-05:00 INFO 17544 --- [main] org.mongodb.driver.client : MongoClient with metadata {"driver": {"name": "mongo-java-driver"}, "sync": true}
2024-09-14T21:51:17.683-05:00 INFO 17544 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8087 (http) with context path ''
2024-09-14T21:51:17.690-05:00 INFO 17544 --- [main] c.c.ejemplo.S07DevEjemplo03Application : Started S07DevEjemplo03Application in 3.773 seconds (process running for 3.773)
2024-09-14T21:51:18.479-05:00 INFO 17544 --- [localhost:27017] org.mongodb.driver.cluster : Monitor thread successfully connected to server with description Server[localhost:27017]

```

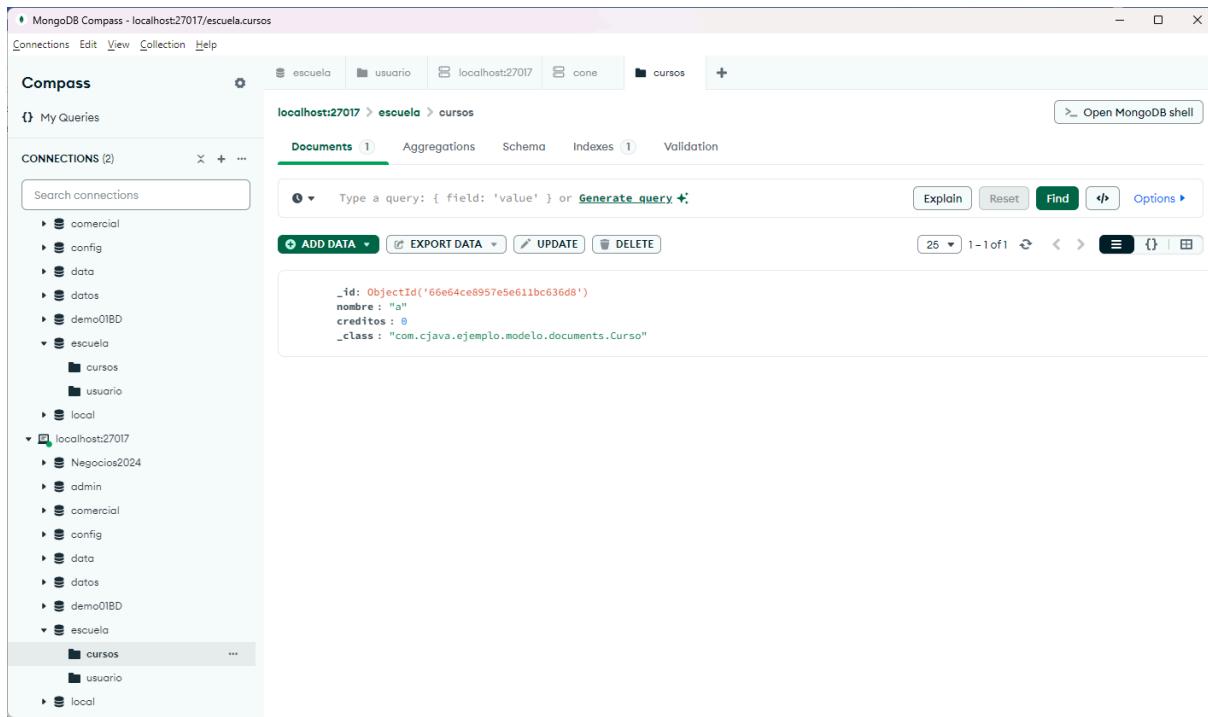
15.- Insertar un registro

CODIGO	NOMBRES	CREDITOS	EDITAR	ELIMINAR
1	Programmer	5	<button>editar</button>	<button>eliminar</button>
2	Developer	5	<button>editar</button>	<button>eliminar</button>
3	Expert	5	<button>editar</button>	<button>eliminar</button>

Nombre	DEMO
Creditos	0
<button>Crear Curso</button>	

CODIGO	NOMBRES	CREDITOS	EDITAR	ELIMINAR
1	Programmer	5	<button>editar</button>	<button>eliminar</button>
2	Developer	5	<button>editar</button>	<button>eliminar</button>
3	Expert	5	<button>editar</button>	<button>eliminar</button>
5	DEMOSTRACION	0	<button>editar</button>	<button>eliminar</button>

16.- Revisar la Base de Datos



The screenshot shows the MongoDB Compass interface. On the left, the connection tree displays two connections: 'escuela' and 'localhost:27017'. Under 'escuela', there are collections 'cursos' and 'usuario'. Under 'localhost:27017', there are collections 'Negocios2024', 'admin', 'comercial', 'config', 'data', 'datos', 'demoQIBD', 'escuela', 'cursos', 'usuario', and 'local'. The main panel shows the 'cursos' collection under the 'escuela' database. It has 1 document listed. The document details are:

```
_id: ObjectId('66e64ce8957e5e611bc636d8')
nombre : "a"
creditos : 0
__class : "com.cjava.ejemplo.modelo.documents.Curso"
```

Ejercicios de aplicación

El proyecto de la evaluación 02, realizarlo con MongoDB.

Conclusiones:

Indicar 5 conclusiones que llegó después de los temas tratados de manera práctica en este laboratorio.
