



GUÍA PRÁCTICA

1. Datos Generales

Carrera:	Tecnología Superior en Desarrollo de Software
Período académico:	Abril 2023 – Agosto 2023
Asignatura:	Tendencias actuales de programacion
Unidad N°:	4
Tema:	Integración de una base de datos no relacional en un sistema, microservicio Springboot Mongodb
Ciclo-Paralelo:	M5b
Fecha de inicio de la Unidad:	20/06/2023
Fecha de fin de la Unidad	17/07/2023
Práctica N°:	3
Horas:	20
Docente:	Ing. Diego Cale MgSc

2. Contenido

2.1 Introducción

El incremento de grandes cantidades de información plantea un nuevo desafío para las tecnologías, los sistemas informáticos de hoy deben manejar un proceso de almacenamiento más versátil adaptable a todo tipo de formato y estructura con alta disponibilidad en el manejo de datos, información no estructurada y manejo de sesiones todo con un buen rendimiento. Mongodb es una base de datos que almacena información como Documentos, al tener una estructura JSON, permite que la integración con servicios web sea sencilla. De la misma manera el framework Springboot permite desarrollar microservicios de una manera sencilla e intuitiva, al tener integrado repositorios para el almacenamiento de datos en Mongodb, y otras bases de datos NoSQL es una herramienta que actualmente es preferida para el desarrollo de Apis de servicios web.

2.2 Objetivo de la Guía

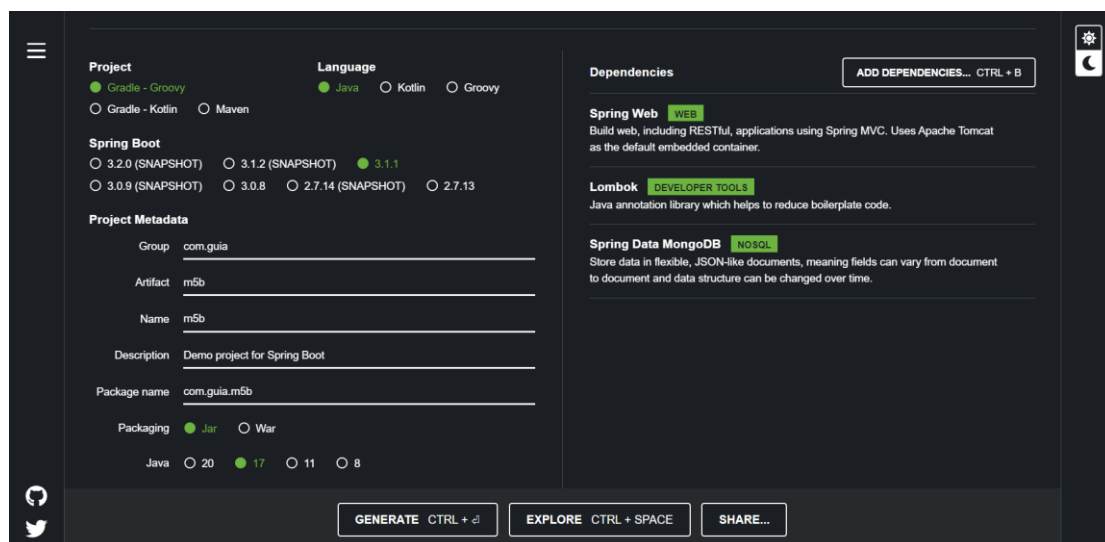
Desarrollar un API con microservicios para almacenar datos en una base de datos NoSQL, ejemplo con Springboot y MongoDB.

2.3 Materiales, herramientas, equipos y software

- Equipos de computación
- Eclipse, Springboot Tool Suite o Netbeans
- Navegador Web
- Material Guía (Talleres, ejercicios prácticos)

2.4 Procedimiento

Paso 1: Spring Initialzer

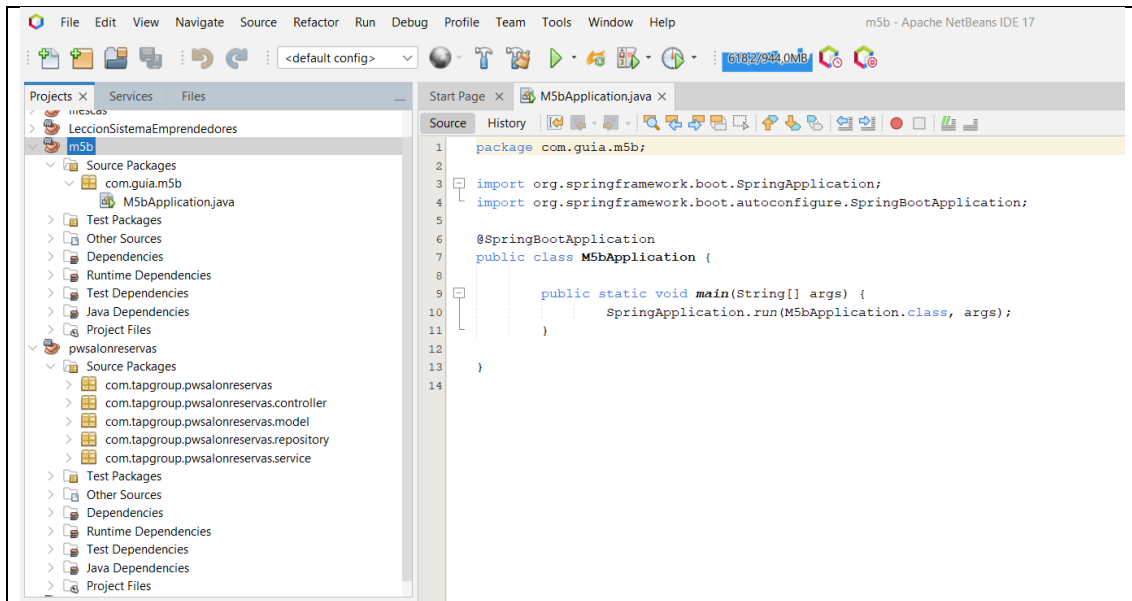


The screenshot shows the Spring Initializer web application interface. It is divided into several sections:

- Project:** Includes radio buttons for **Gradle - Groovy** (selected), **Gradle - Kotlin**, and **Maven**.
- Language:** Includes radio buttons for **Java** (selected), **Kotlin**, and **Groovy**.
- Spring Boot:** Includes radio buttons for versions: **3.2.0 (SNAPSHOT)**, **3.1.2 (SNAPSHOT)**, **3.1.1** (selected), **3.0.9 (SNAPSHOT)**, **3.0.8**, **2.7.14 (SNAPSHOT)**, and **2.7.13**.
- Project Metadata:** Includes input fields for **Group** (com.guia), **Artifact** (m5b), **Name** (m5b), **Description** (Demo project for Spring Boot), and **Package name** (com.guia.m5b).
- Packaging:** Includes radio buttons for **Jar** (selected) and **War**.
- Java:** Includes radio buttons for versions: **20**, **17** (selected), **11**, and **8**.
- Dependencies:** Includes a button **ADD DEPENDENCIES... CTRL + B** and a list of dependencies:
 - Spring Web** (WEB): Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.
 - Lombok** (DEVELOPER TOOLS): Java annotation library which helps to reduce boilerplate code.
 - Spring Data MongoDB** (NOSQL): Store data in flexible, JSON-like documents, meaning fields can vary from document to document and data structure can be changed over time.

At the bottom, there are three buttons: **GENERATE CTRL + G**, **EXPLORE CTRL + SPACE**, and **SHARE...**

Paso 2: modelo y repositorio



```
13
14 /**
15  *
16  * @author LENOVO
17  */
18 @Document(collection = "Estudiante")
19 @Data
20 public class Estudiante {
21
22     @JsonProperty("_id")
23     private int _id;
24     private String nombre;
25     private int numero_estudiante;
26     private String correo;
27     private List<Curso> cursosList;
28     private Depto gpa;
29
30
31 }
32
```

Paso 3: Servicio al estudiante

Paso 4: controlador REST

```
Source History
1 package controller;
2
3 import java.util.List;
4 import modelo.Estudiante;
5 import org.springframework.beans.factory.annotation.Autowired;
6 import org.springframework.dao.DataAccessException;
7 import org.springframework.http.HttpStatus;
8 import org.springframework.http.ResponseEntity;
9 import org.springframework.web.bind.annotation.DeleteMapping;
10 import org.springframework.web.bind.annotation.GetMapping;
11 import org.springframework.web.bind.annotation.PathVariable;
12 import org.springframework.web.bind.annotation.PostMapping;
13 import org.springframework.web.bind.annotation.PutMapping;
14 import org.springframework.web.bind.annotation.RequestBody;
15 import org.springframework.web.bind.annotation.RequestMapping;
16 import org.springframework.web.bind.annotation.RestController;
17 import service.EstudianteServiceImpl;
18
19 @RestController
20 @RequestMapping("/api/estudiante")
21 public class EstudianteController {
22
23     @Autowired
24     EstudianteServiceImpl servEst;
25
26     @GetMapping("/listar")
27     public ResponseEntity<List<Estudiante>> listarEstudiante() {
28         return new ResponseEntity<>(servEst.findByAll(), HttpStatus.OK);
29     }
30
31     @PostMapping("/crear")
32     public ResponseEntity<Estudiante> crearEstudiante(@RequestBody Estudiante p) {
33         return new ResponseEntity<>(servEst.save(p), HttpStatus.CREATED);
34     }
35 }
```

Paso 5: Configuración de la base de datos

```
...va EstudianteController.java application.properties
Source History
1 #server
2 server.port = 8080
3
4 #mongodb
5 spring.data.mongodb.host = localhost
6 spring.data.mongodb.port = 27017
7 spring.data.mongodb.database = guia
8
9
```

Ejecución del servicio y pruebas

```

{"t":{"$date":"2023-07-17T23:04:39.769-05:00"},"s":"I", "c":"CONTROL", "id":23285, "ctx":{"thread1","msg":"Automatically disabling TLS 1.0, to force-enable TLS 1.0 s
specify --sslDisabledProtocols 'none'"}}
{"t":{"$date":"2023-07-17T23:04:41.920-05:00"},"s":"I", "c":"NETWORK", "id":4915701, "ctx":{"thread1","msg":"Initialized wire specification","attr":{"spec":{"incomingI
ternalClient":{"minWireVersion":0,"maxWireVersion":17},"incomingInternalClient":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"minWireVersion":6,"maxWireVersion
":17},"isInternalClient":true}}}}
{"t":{"$date":"2023-07-17T23:04:41.932-05:00"},"s":"I", "c":"NETWORK", "id":4648602, "ctx":{"thread1","msg":"Implicit TCP FastOpen in use."}}
{"t":{"$date":"2023-07-17T23:04:41.985-05:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":{"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"servi
ce":"TenantMigrationDonorService","namespace":"config.tenantMigrationDonors"}}}
{"t":{"$date":"2023-07-17T23:04:41.986-05:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":{"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"servi
ce":"TenantMigrationRecipientService","namespace":"config.tenantMigrationRecipients"}}}
{"t":{"$date":"2023-07-17T23:04:41.986-05:00"},"s":"I", "c":"REPL", "id":5123008, "ctx":{"thread1","msg":"Successfully registered PrimaryOnlyService","attr":{"servi
ce":"ShardSplitDonorService","namespace":"config.tenantSplitDonors"}}}
{"t":{"$date":"2023-07-17T23:04:41.987-05:00"},"s":"I", "c":"CONTROL", "id":5945603, "ctx":{"thread1","msg":"Multi threading initialized"}}
{"t":{"$date":"2023-07-17T23:04:41.993-05:00"},"s":"I", "c":"CONTROL", "id":4615611, "ctx":{"initandlisten","msg":"MongoDB starting","attr":{"pid":20864,"port":27017,"
dbPath":"c:/data/db/","architecture":"64-bit","host":"DESKTOP-HDQGHVQ"}}}
{"t":{"$date":"2023-07-17T23:04:41.993-05:00"},"s":"I", "c":"CONTROL", "id":23398, "ctx":{"initandlisten","msg":"Target operating system minimum version","attr":{"ta
getMinOs":"Windows 7/Windows Server 2008 R2"}}}
{"t":{"$date":"2023-07-17T23:04:41.994-05:00"},"s":"I", "c":"CONTROL", "id":23403, "ctx":{"initandlisten","msg":"Build Info","attr":{"buildInfo":{"version":"6.0.7","
distmod":"x86_64","distarch":"x86_64","target_arch":"x86_64","target_os":"windows","target_cpu":"x86_64","modules":["allocator":"tcmalloc","environment":{"distmod":"windows","distarch":"x86_64","target_arch":"x86_64"}]}}}}}
{"t":{"$date":"2023-07-17T23:04:41.994-05:00"},"s":"I", "c":"CONTROL", "id":51765, "ctx":{"initandlisten","msg":"Operating System","attr":{"os":{"name":"Microsoft Wi
ndows 10","version":"10.0 (build 22000)}}}}
{"t":{"$date":"2023-07-17T23:04:41.994-05:00"},"s":"I", "c":"CONTROL", "id":21951, "ctx":{"initandlisten","msg":"Options set by command line","attr":{"options":{}}}}
{"t":{"$date":"2023-07-17T23:04:42.013-05:00"},"s":"E", "c":"CONTROL", "id":20557, "ctx":{"initandlisten","msg":"DBException in initAndlisten, terminating","attr":{"
error":"NonExistentPath: Data directory C:\\data\\db\\ not found. Create the missing directory or specify another path using (1) the --dbpath command line option, or (2)
by adding the 'storage.dbPath' option in the configuration file."}}}}
{"t":{"$date":"2023-07-17T23:04:42.015-05:00"},"s":"I", "c":"REPL", "id":4784900, "ctx":{"initandlisten","msg":"Stepping down the ReplicationCoordinator for shutdown
","attr":{"waitTimeMillis":15000}}}}
{"t":{"$date":"2023-07-17T23:04:42.022-05:00"},"s":"I", "c":"REPL", "id":4794602, "ctx":{"initandlisten","msg":"Attempting to enter quiesce mode"}}
{"t":{"$date":"2023-07-17T23:04:42.023-05:00"},"s":"I", "c":"-", "id":6371601, "ctx":{"initandlisten","msg":"Shutting down the File Crud thread pool"}}
{"t":{"$date":"2023-07-17T23:04:42.024-05:00"},"s":"I", "c":"COMMAND", "id":4784901, "ctx":{"initandlisten","msg":"Shutting down the MirrorMaestro"}}
{"t":{"$date":"2023-07-17T23:04:42.025-05:00"},"s":"I", "c":"SHARDING", "id":4784902, "ctx":{"initandlisten","msg":"Shutting down the WaitForMajorityService"}}
{"t":{"$date":"2023-07-17T23:04:42.025-05:00"},"s":"I", "c":"NETWORK", "id":20562, "ctx":{"initandlisten","msg":"Shutdown: going to close listening sockets"}}
{"t":{"$date":"2023-07-17T23:04:42.026-05:00"},"s":"I", "c":"NETWORK", "id":4784905, "ctx":{"initandlisten","msg":"Shutting down the global connection pool"}}
{"t":{"$date":"2023-07-17T23:04:42.026-05:00"},"s":"I", "c":"CONTROL", "id":4784906, "ctx":{"initandlisten","msg":"Shutting down the FlowControlTicketHolder"}}
{"t":{"$date":"2023-07-17T23:04:42.027-05:00"},"s":"I", "c":"-", "id":20520, "ctx":{"initandlisten","msg":"Stopping further Flow Control ticket acquisitions."}}

```

2.5 Resultados esperados

Conocer la integración de una base de datos no relacional con un aplicativo.

Desarrollar un aplicativo API, de servicios web para interactuar con Springboot y MongoDB.

Eliminar un alumno Después de realizar los pasos anteriores, debería ver dos registros en la base de datos como se muestra a continuación. La colección de estudiantes

Realizar pruebas con los servicios web utilizando Postman.

2.6 Bibliografía

Descripción en norma APA
Medium. 2020. Building A REST Service With Spring Boot And MongoDB. [online] Available at: < https://medium.com/javarevisited/building-a-rest-service-with-spring-boot-andmongodbpart-1-2de01e4f434d > [Accessed 13 feb 2020].
Treehouse.github.io. 2020. Installing MongoDB On Windows. [online] Available at: [Accessed 13 feb 2020].

3 Firmas de Responsabilidad

ESTUDIANTE	DOCENTE	DIRECTORA DE CARRERA
------------	---------	----------------------

	Nombre:	Nombre: Ing. Jéssica Herrera Urgilés, Mgtr.
Firma	Firma	Firma
/Fecha:17/072023	Fecha: 17/072023	Fecha: 17/072023