



Formando líderes para la construcción de un nuevo país en paz

LABORATORIO CRUD

Estudiantes:

Wilber Santiago Barajas Cordero 1093591966

Yenderson Josué Rangel Martínez 1127045112

DESARROLLO ORIENTADO A PLATAFORMAS

Grupo: BR

FANNY CASADIEGO CHIQUILLO

26 de noviembre del 2025



INDICE

1. Laboratorio	3
2. Solución Taller	6
3. Documento corto respondiendo	11



1. Laboratorio

Gestión de Estudiantes con Spring Boot

Construcción de un servicio REST para la gestión de estudiantes usando Spring Boot

En este laboratorio, debe implementar un **servicio web REST** utilizando **Spring Boot**, con el fin de gestionar un conjunto de estudiantes en una aplicación backend.

El propósito es reforzar conceptos de controladores REST, modelos de datos, servicios de negocio y manejo de colecciones en memoria.

El servicio permitirá realizar las operaciones básicas de un CRUD (**Crear, Leer, Actualizar y Eliminar**) sin utilizar todavía bases de datos externas.

Los datos serán almacenados en una colección interna para facilitar la comprensión de la arquitectura inicial.

Contexto del ejercicio:

La Universidad desea implementar un módulo básico para registrar estudiantes en un prototipo de plataforma académica. Por ahora, solo se necesita un servicio backend que permita registrar estudiantes y consultar su información a través de rutas HTTP.

Este laboratorio sirve como base para futuros módulos como matrículas, asignaturas o notas.

El objetivo principal es que el estudiante comprenda el flujo entre:

- **Controlador REST**



- **Servicio de lógica de negocio**
- **Modelo de datos**
- **Uso de rutas HTTP (GET, POST, PUT, DELETE)**

Para controlar el modelo se deben usar:

- Crear controladores REST usando `@RestController` y `@RequestMapping`.

`@GetMapping`, `@PostMapping`, `@PutMapping`, `@DeleteMapping`, `@Service`, Manejo de listas en memoria (`ArrayList`).

- Implementar modelos de datos en Java.
- Separar responsabilidades mediante un servicio de negocio (`@Service`).
- Crear endpoints que reciban y respondan datos en formato JSON.
- Usar Postman o navegador para consumir servicios REST.
- Comprender el funcionamiento básico de un CRUD antes de integrar bases de datos.

Requerimientos funcionales:

El sistema debe permitir:

1. **Listar todos los estudiantes (GET).**
2. **Buscar un estudiante por ID (GET).**
3. **Registrar un nuevo estudiante (POST).**
4. **Actualizar los datos de un estudiante existente (PUT).**
5. **Eliminar un estudiante por ID (DELETE).**



Cada estudiante debe contener como mínimo:

- id (entero)
- nombre (cadena de texto)
- edad (entero)

Los datos iniciales deben incluir al menos dos estudiantes de ejemplo cargados en memoria.

OJO:

- Código fuente completo del proyecto con las siguientes capas:

model

service

Controller



2. Solución Taller

- Listar Estudiantes

The screenshot shows a REST client interface with a GET request to `http://localhost:8080/est`. The response is a JSON object with the following structure:

```

{
  "metadata": [
    {
      "Fecha": "2025-11-28",
      "Codigo": "200",
      "Estado": "OK"
    }
  ],
  "estudianteResponse": {
    "estudiantes": [
      {
        "id": 1,
        "nombre": "Ana",
        "edad": 20
      },
      {
        "id": 2,
        "nombre": "Carlos",
        "edad": 22
      },
      {
        "id": 3,
        "nombre": "Maria",
        "edad": 19
      }
    ]
  }
}

```

Below the response, the terminal shows the following log messages:

```

2025-11-28T12:11:55.423-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] o.a.c.c.C.[Tomcat].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-11-28T12:11:55.423-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-11-28T12:11:55.423-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 0 ms
2025-11-28T12:11:55.445-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] c.c.c.dao.service.EstudianteServiceImpl : Entrando al método de listar todos los estudiantes
2025-11-28T12:11:55.447-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] c.c.c.dao.service.EstudianteServiceImpl : Devolviendo respuesta al cliente

```



- Buscar por ID

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda

Q Laboratorio-CRUD

TC New Request X

GET http://localhost:8080/est/1 Send

Query Headers Auth Body Tests Pre Run

Query Parameters

parameter value

Status: 200 OK Size: 139 Bytes Time: 90 ms

Response Headers Cookies Results Docs

```
1 {
2   "metadata": {
3     {
4       "Fecha": "2025-11-28",
5       "Codigo": "200",
6       "Estado": "OK"
7     }
8   },
9   "estudianteResponse": {
10    "estudiantes": [
11      {
12        "id": 1,
13        "nombre": "Ana",
14        "edad": 20
15      }
16    ]
17  }
18 }
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\Usuario\Downloads\Laboratorio-CRUD> & 'C:\Program Files\Java\jdk-21\bin\java.exe' '@C:\Users\Usuario\AppData\Local\Temp\cp_5ogpcukmg7ffh4pySzi9ev2.argfile' 'com.crud.crud.dao.Application' at org.apache.tomcat.util.threads.TaskThread\$WrappingRunnable.run(TaskThread.java:63) -[tomcat-embed-core-10.1.43.jar:10.1.43] at java.base/java.lang.Thread.run(Thread.java:1583) -[na:na]

```
2025-11-28T12:11:55.423-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dispatcherServlet'
2025-11-28T12:11:55.423-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2025-11-28T12:11:55.423-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] o.s.web.servlet.DispatcherServlet : Completed initialization in 0 ms
2025-11-28T12:11:55.445-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] c.c.c.dao.service.EstudianteServiceImpl : Entrando al método de listar todos los estudiantes
2025-11-28T12:11:55.447-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-2] c.c.c.dao.service.EstudianteServiceImpl : Devolviendo respuesta al cliente
2025-11-28T12:51:32.362-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-3] c.c.c.dao.service.EstudianteServiceImpl : Entrando al método de obtener estudiante por id
2025-11-28T12:51:32.365-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-3] c.c.c.dao.service.EstudianteServiceImpl : Intentando recuperar estudiante con id 1
2025-11-28T12:51:32.369-05:00 INFO 9372 --- [crud.dao] [nio-8080-exec-3] c.c.c.dao.service.EstudianteServiceImpl : Devolviendo respuesta al cliente
```

Java: Ready



- Crear Estudiante

```
POST http://localhost:8080/est
{
  "id": 4,
  "nombre": "Johander",
  "edad": 21
}
```

```
{
  "metadata": {
    "fecha": "2025-11-28",
    "codigo": "200",
    "estado": "OK"
  },
  "estudianteResponse": {
    "estudiantes": [
      {
        "id": 4,
        "nombre": "Johander",
        "edad": 21
      }
    ]
  }
}
```

```
GET http://localhost:8080/est/4
```

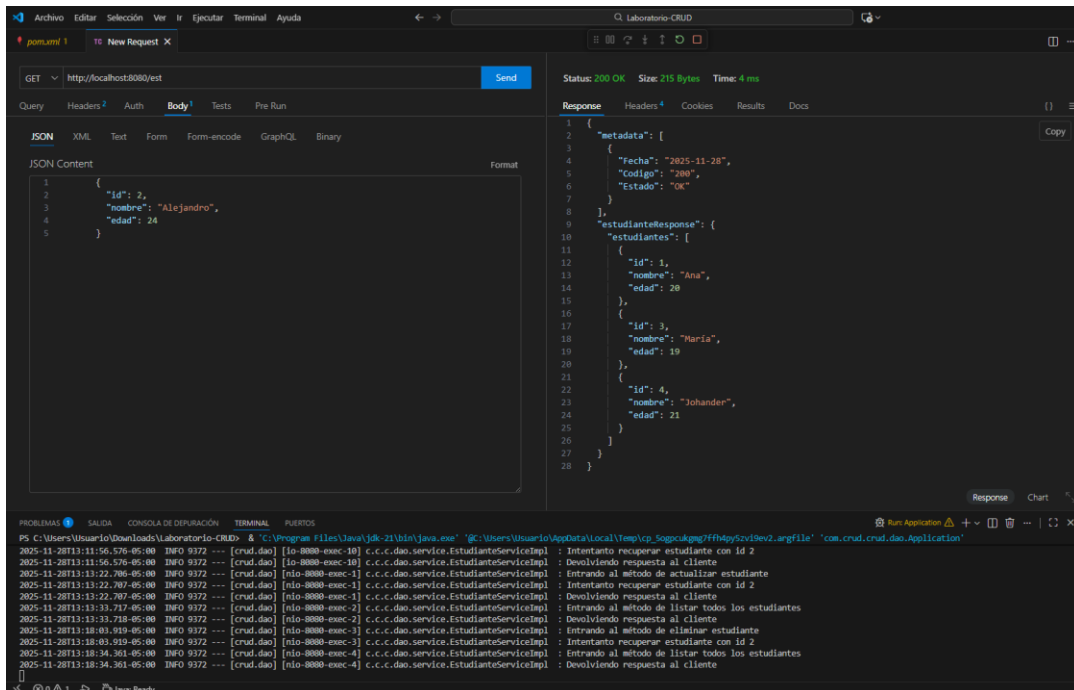
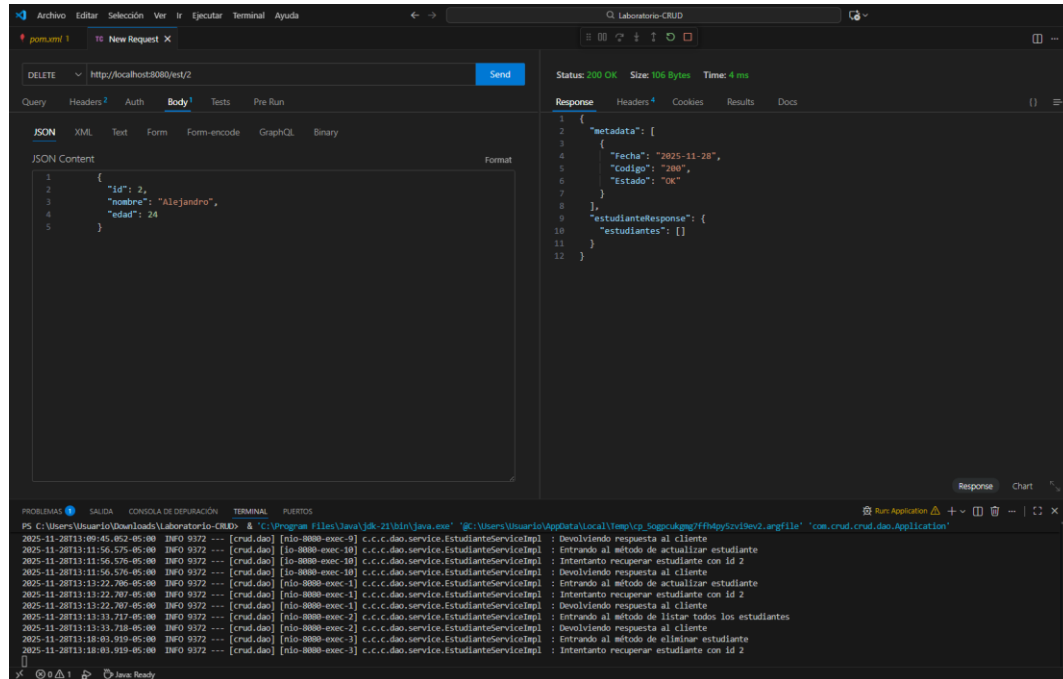
```
{
  "metadata": {
    "fecha": "2025-11-28",
    "codigo": "200",
    "estado": "OK"
  },
  "estudianteResponse": {
    "estudiantes": [
      {
        "id": 4,
        "nombre": "Johander",
        "edad": 21
      }
    ]
  }
}
```




- Actualizar Estudiante



- Eliminar Estudiante





3. Documento corto respondiendo

- ¿Qué ventajas y desventajas tiene usar listas en memoria?

RTA:// Usar listas en memoria permite desarrollar muy rápido porque no requiere configurar una base de datos y facilita las pruebas iniciales del CRUD. Además, mientras el proyecto está en desarrollo temprano, trabajar con listas hace más sencillo entender la lógica sin preocuparse por transacciones ni consultas complejas.

Sin embargo, también tiene desventajas importantes ya que los datos se pierden cada vez que se reinicia la aplicación, el tamaño está limitado por la memoria del servidor y no existe ningún mecanismo automático que garantice integridad, unicidad o validaciones. Tampoco es adecuado para manejar consultas avanzadas o grandes volúmenes de información. Es una solución útil solo para aprender o prototipar, pero no para un sistema real.

- ¿Qué ocurriría si hubiese múltiples usuarios accediendo al sistema al mismo tiempo?

RTA:// Si varios usuarios acceden al sistema simultáneamente mientras se usan listas en memoria, se pueden presentar problemas de concurrencia. Dado que la lista es compartida por todos los hilos, dos usuarios podrían intentar modificarla al mismo tiempo, lo que puede generar datos inconsistentes, actualizaciones sobreescritas o información corrupta. Además, al no tener aislamiento ni transacciones, los usuarios pueden leer datos en un estado intermedio o recibir resultados distintos dependiendo del momento exacto en que la lista esté siendo modificada.

- ¿Qué mejorarías si usaras una base de datos real?

RTA:// Si se reemplaza la lista en memoria por una base de datos real, se obtiene persistencia, por lo que los datos ya no se pierden al reiniciar el sistema. Además, el motor de la base de datos maneja la concurrencia con transacciones, bloqueos y mecanismos que garantizan que varios usuarios puedan leer y modificar la información sin causar errores. También se habilita la posibilidad de hacer consultas avanzadas, filtros, ordenamientos y relaciones entre tablas, algo imposible o muy limitado con listas. La integridad de los datos mejora gracias a claves primarias, foráneas y restricciones, y el sistema se vuelve más escalable y apto para crecer, tanto en usuarios como en volumen de información.