Ejercicio 2(tarea 2)

- Implementar el Servicio de CalificacionesDto

```
package mx.unam.dgtic.alumnos_rest.servicios.v2.interfaces;

import mx.unam.dgtic.alumnos_rest.dtos.CalificacionDto;

import java.util.List;

public interface CalificacionDtoService { 6 usages 1 implementation

List<CalificacionDto > findAll(); 1 usage 1 implementation

CalificacionDto findById(Long id); 1 implementation

CalificacionDto save(CalificacionDto calificacionDto); 1 usage 1 implementation

CalificacionDto update(Long id, CalificacionDto calificacionDto); 1 usage 1 implementation

CalificacionDto delete(Long id); 1 usage 1 implementation

List<CalificacionDto > getPagina(int page, int size, String direction, String sort);

}
```

- Rest Controller completo para Calificaciones Dto

```
import mx.unam.dgtic.alumnos_rest.servicios.v2.interfaces.CalificacionDtoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
   public ResponseEntity<List<CalificacionDto>> getAll() {
   @GetMapping("/{id}") no usages
   public ResponseEntity<CalificacionDto> getById(@PathVariable Long id) {
       return ResponseEntity.ok(service.findById(id));
   @PostMapping("/") no usages
   public ResponseEntity<CalificacionDto> save(@RequestBody CalificacionDto dto) {
    @PutMapping("/{id}") no usages
    public ResponseEntity<CalificacionDto> update(@PathVariable Long id, @RequestBody CalificacionDto dto) {
    public ResponseEntity<CalificacionDto> delete(@PathVariable Long id) {
```

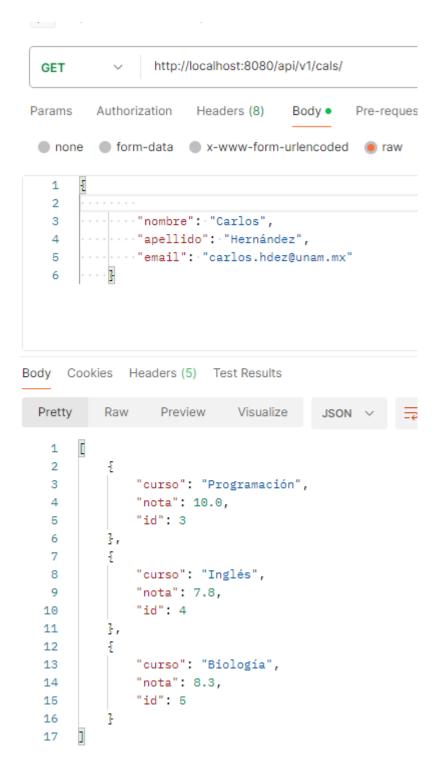
- Implementar paginado para CalificacionesDto.

```
package mx.unam.dgtic.alumnos_rest.controladores.v2;
import mx.unam.dgtic.alumnos_rest.dtos.CalificacionDto;
import mx.unam.dgtic.alumnos_rest.servicios.v2.interfaces.CalificacionDtoService;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.http.ResponseEntity;
import org.springframework.web.bind.annotation.*;
import java.util.List;
@RestController no usages
@RequestMapping("/api/v2/paginable/calificaciones")
   @Autowired 1 usage
    private CalificacionDtoService service;
    public ResponseEntity<List<CalificacionDto>> getPagina(
            @RequestParam(defaultValue = "1") int page,
            @RequestParam(defaultValue = "5") int size,
            @RequestParam(defaultValue = "asc") String direction,
            @RequestParam(defaultValue = "id") String sort
       return ResponseEntity.ok(service.getPagina(page, size, direction, sort));
```

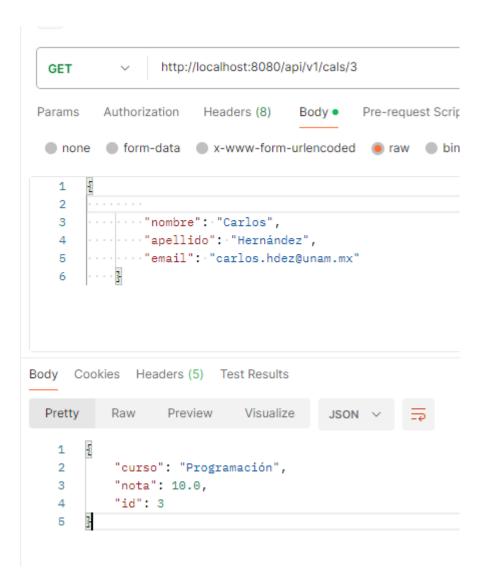
Postman:

CalificacionRestController

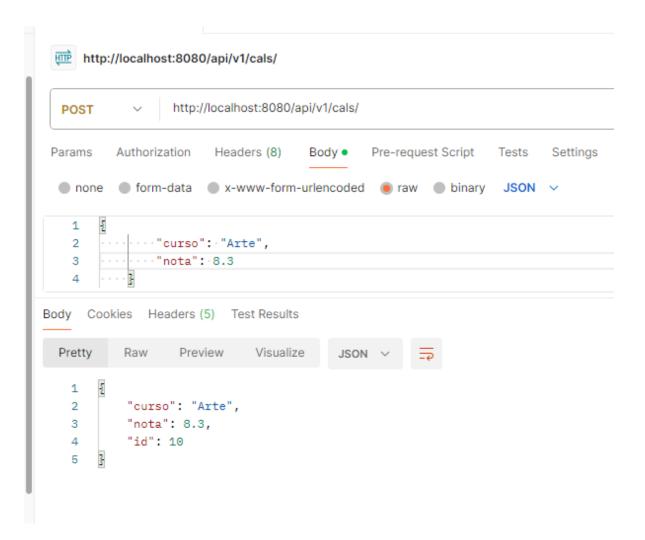
Metodo GET:



Metodo GET{id}:



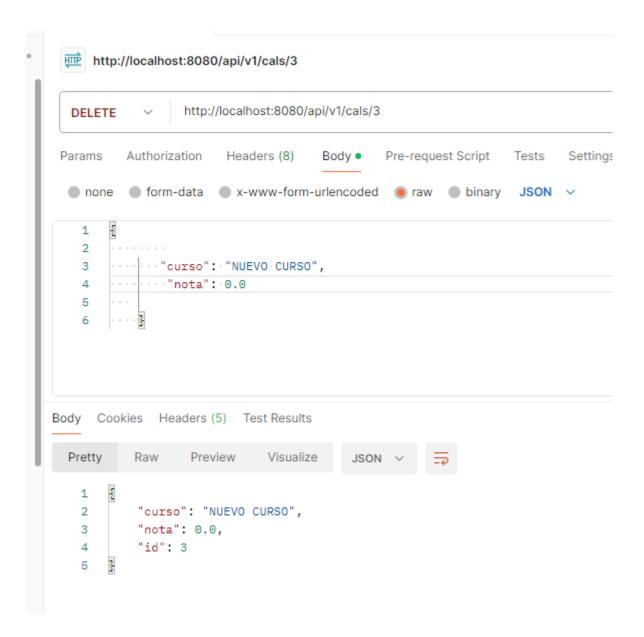
Metodo POST:



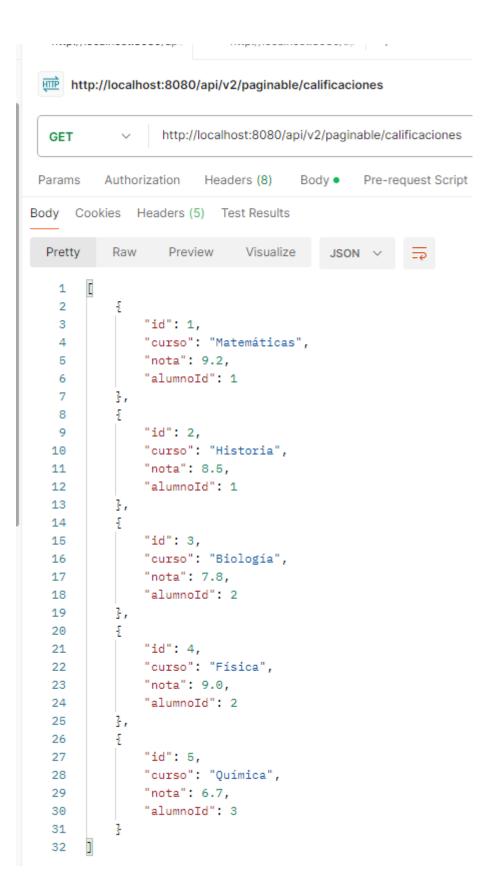
Metodo PUT:

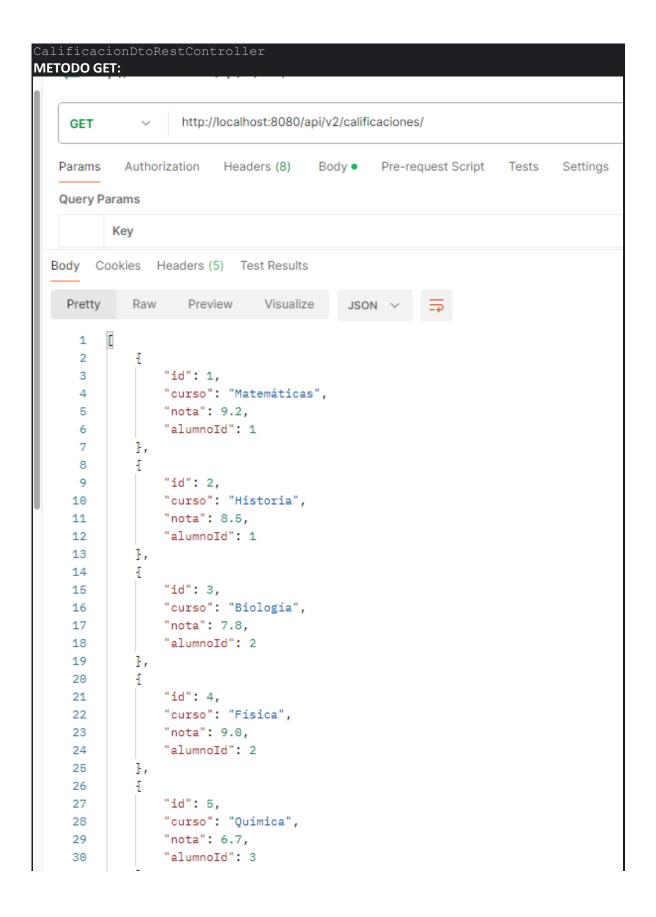


Metodo DELETE:

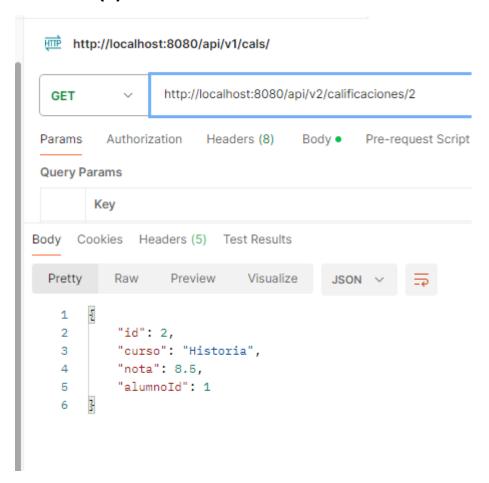


PAGINACION:

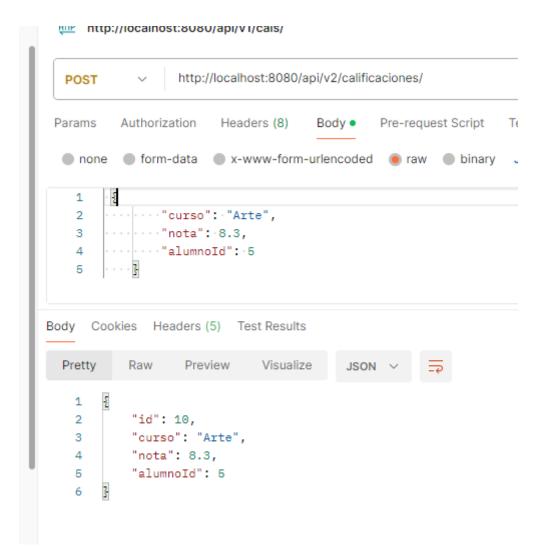




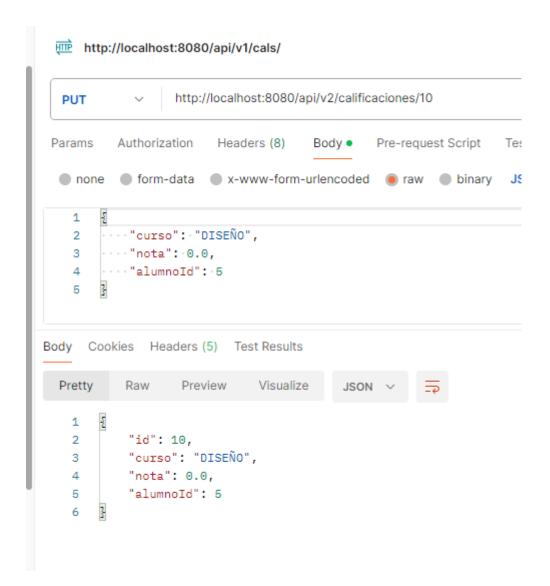
METODO GET{id}:



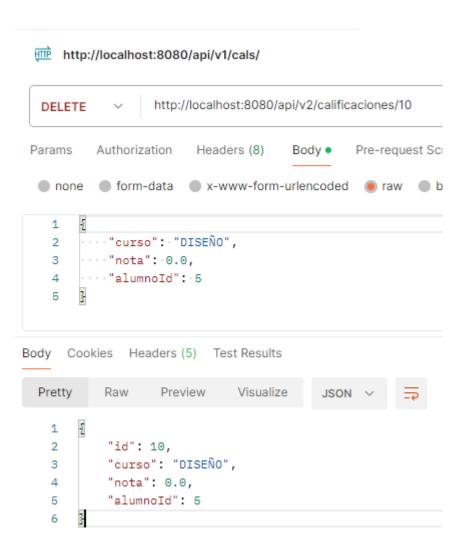
METODO POST:



METODO PUT:



METODO DELETE:



http://localhost:8080/api/v1/cals/

```
http://localhost:8080/api/v2/calificaciones/
  GET
          Authorization
                        Headers (8)
                                      Body •
                                               Pre-request Script
 Params
                                                                  Tests
                                                                          Sŧ
  none form-data x-www-form-urlencoded raw binary
                                                                   JSON V
    1
        ····"curso": "DISEÑO",
    2
        ····"nota": 0.0,
    3
        ····"alumnoId": 5
    4
    5
Body Cookies Headers (5) Test Results
  Pretty
           Raw
                   Preview
                               Visualize
                                           JSON V
   30
                "alumnoId": 3
   31
            },
   32
   33
                "id": 6,
               "curso": "Literatura",
   34
                "nota": 8.0,
   35
                "alumnoId": 4
   36
   37
            },
            £
   38
   39
                "id": 7,
                "curso": "Matemáticas",
   40
                "nota": 9.5,
   41
   42
                "alumnoId": 5
   43
            ₹,
   44
                "id": 8,
   45
                "curso": "Inglés",
   46
                "nota": 7.9,
   47
                "alumnoId": 3
   48
   49
            },
   50
                "id": 9,
   51
                "curso": "Computación",
   52
                "nota": 10.0,
   53
                "alumnoId": 4
   54
   55
   56
```