# TECNOLOGÍA SUPERIOR EN DESARROLLO DE SOFTWARE

**TENDENCIAS ACTUALES DE LA PROGRAMACIÓN**

**Ciclo: M5B**

**Nombres Completos: Christian Javier Illescas Cuenca**

| CRITERIO | DESCRIPCIÓN | EVIDENCIA(FOTOCOPIA) |
|---|---|---|
| **Define la Entidad y un Repositorio** | Crea la entidad con los atributos validados correctamente |  |

```java
import java.util.List;

/**
 *
 * @author LENOVO
 */
public class Profesor {

    private int idprofesor;
    private String nombre;
    private String apellido;
    private String celular;
    private String correo;
    private String direccion;

    private List<Asignatura> asigantura;
    private List<Carrera> carrera;
    private List<Ciclo> ciclo;
    private List<Periodo> periodo;

}
```

```java
/**
 *
 * @author LENOVO
 */
public class Periodo {

    private int idperiodo;
    private String periodo;
    private int year;

}
```

```java
 * @author LENOVO
 */
public class Ciclo {

    private int idciclo;
    private String nombre;
    private String descripcion;

}
```

```
9    * @author LENOVO
0    */
1   public class Carrera {
2
     private int idcarrera;
     private String nombre;
     private String descripcion;
6
7
8
9
0   }
```

```
 * @author LENOVO
 */
public class Asignatura {

    private int idasignatura;
    private String nombre;
    private int  n_horas;
    private int n_horas_practicas;
    private int n_horas_autonomas;

}
```

| Define y extiende la interfaz del Repositorio correctamente | |

```java
Source   History
1  /*
2   * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to c
3   * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Interface.java to edit th
4   */
5  package com.tendencias.examenm5billescas.repositorio;
6
7  import com.tendencias.examenm5billescas.modelo.Profesor;
8  import org.springframework.data.mongodb.repository.MongoRepository;
9
10
11 public interface ClaseRepository extends MongoRepository<Profesor, Long>{
12
13 }
14
```

| Define la Clase Controladora | Crea la clase controlador con los métodos GET, POST, PUT, DELETE correctamente | |
|---|---|---|
| | | ```
Source  History
28  @RestController
29  @RequestMapping("/api/profesor")
30  public class ProfesorController {
31
32      @Autowired
33      Clase1ServiceImpl servProf;
34
35      @GetMapping("/listar")
36      public ResponseEntity<List<Profesor>> listarProfesor() {
37          return new ResponseEntity<>(servProf.findByAll(), HttpStatus.OK);
38      }
39
40      @PostMapping("/crear")
41      public ResponseEntity<Profesor> crearProfesor(@RequestBody Profesor p) {
42          return new ResponseEntity<>(servProf.save(p), HttpStatus.CREATED);
43      }
44
45      @PutMapping("/actualizar/{id}")
46      public ResponseEntity<Profesor> actualizarProfesor(@PathVariable Long id, @Req
47          Profesor profesor = servProf.findById(id);
48          if (profesor == null) {
49              return new ResponseEntity<>(HttpStatus.NOT FOUND);
50          } else {
51
52              p.set_id(profesor.get_id());
53              return new ResponseEntity<>(servProf.save(p), HttpStatus.OK);
54
55          }
56      }
57
58      @DeleteMapping("/eliminar/{id}")
59      public ResponseEntity<Profesor> eliminarProducto(@PathVariable Long id) {
60          servProf.delete(id);
``` |
| **Crea el microservicio** | Crea el microservicio e implementa los métodos de las operaciones CRUD correctamente |  |

Request URL
http://localhost:8000/api/profesor/actualizar/1
Server response
Code    Details
200     Response body
        {
            "periodo": "2023 - 2024",
            "year": 0,
            "_id": 1
        }
        ],
        "carreras": [
        {
            "nombre": "DESARROLLO",
            "descripcion": "Desarrollo de software",
            "_id": 1
        }
        ],
        "ciclos": [
        {
            "nombre": "Cuarto",
            "descripcion": "Cuarto ciclo",
            "_id": 1
        }
        ],
        "asignaturas": [
        {
            "nombre": "Tendencias",
            "n_horas": 23,
            "n_horas_practicas": 42,
            "n_horas_autonomas": 45,
            "_id": 1
        Response headers
        connection: keep-alive
        content-type: application/json
        date: Tue,08 Aug 2023 15:51:43 GMT
        keep-alive: timeout=60

Curl
curl -X 'DELETE' \
  'http://localhost:8000/api/profesor/eliminar/1' \
  -H 'accept: */*'
Request URL
http://localhost:8000/api/profesor/eliminar/1
Server response
Code    Details
200     Response headers
        connection: keep-alive
        content-length: 0
        date: Tue,08 Aug 2023 15:52:10 GMT
        keep-alive: timeout=60
Responses
Code    Description                                                             Lir
200     OK                                                                      No

        Media type
        */*
        Controls Accept header.

        Example Value | Schema
        {
            "nombre": "string",
            "apellido": "string",

| | | |
|---|---|---|
| **Docker, Mongo y consultas** | Realiza la creación de la Bd, y las consultas NoSql respectivas, con el puerto respectivo. | |
| **Scripts de Docker y PlayGround** | Construir los archivos de docker, y Generar el código de consulta MngoDB. | |

| **Documenta las pruebas del microservicio en Swagger** | Documenta las pruebas del microservicio en swagger con los métodos de validación solicitados | |
|---|---|---|

Request URL

`http://localhost:8080/api/profesor/crear`

Server response

| Code | Details |
|---|---|
| 201 Undocumented | Response body |

```
{
  "nombre": "Javier",
  "apellido": "Illescas",
  "correo": "javier@gamil.com",
  "direccion": "el valle",
  "celular": "0987654321",
  "periodos": [
    {
      "periodo": "2023 - 2024",
      "year": 2,
      "_id": 1
    }
  ],
  "carreras": [
    {
      "nombre": "DESARROLLO",
      "descripcion": "Desarrollo de software",
      "_id": 1
    }
  ],
  "ciclos": [
    {
      "nombre": "Quinto",
      "descripcion": "Quinto ciclo",
      "_id": 1
    }
  ],
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Tue, 08 Aug 2023 15:49:51 GMT
```

Curl

```
curl -X 'GET' \
  'http://localhost:8080/api/profesor/listar' \
  -H 'accept: */*'
```

Request URL

`http://localhost:8080/api/profesor/listar`

Server response

| Code | Details |
|---|---|
| 200 | Response body |

```
[
  {
    "nombre": "Javier",
    "apellido": "Illescas",
    "correo": "javier@gamil.com",
    "direccion": "el valle",
    "celular": "0987654321",
    "periodos": [
      {
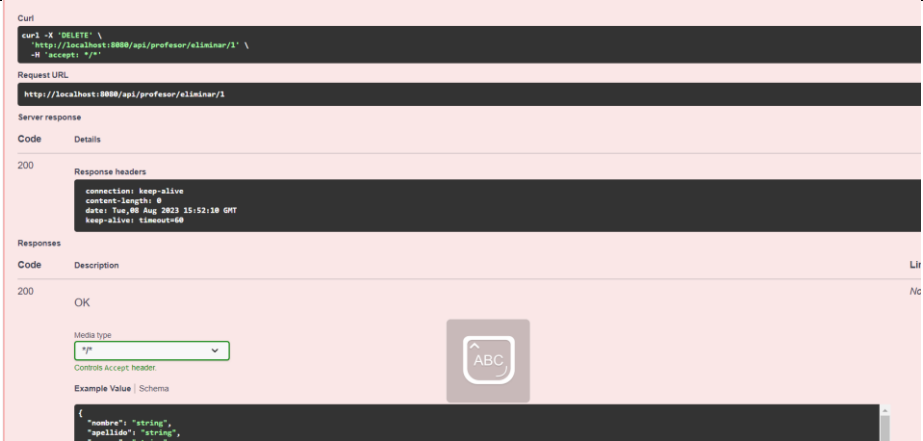        "periodo": "2023 - 2024",
        "year": 2,
        "_id": 1
      }
    ],
    "carreras": [
      {
        "nombre": "DESARROLLO",
        "descripcion": "Desarrollo de software",
        "_id": 1
      }
    ],
    "ciclos": [
      {
        "nombre": "Quinto",
        "descripcion": "Quinto ciclo",
        "_id": 1
```

Request URL

`http://localhost:8080/api/profesor/actualizar/1`

Server response

| Code | Details |
|---|---|
| 200 | Response body |

```
  "periodos": [
    {
      "periodo": "2023 - 2024",
      "year": 2,
      "_id": 1
    }
  ],
  "carreras": [
    {
      "nombre": "DESARROLLO",
      "descripcion": "Desarrollo de software",
      "_id": 1
    }
  ],
  "ciclos": [
    {
      "nombre": "Cuarto",
      "descripcion": "Cuarto ciclo",
      "_id": 1
    }
  ],
  "asignaturas": [
    {
      "nombre": "Tendencias",
      "n_horas": 23,
      "n_horas_practicas": 43,
      "n_horas_autonomas": 45,
      "_id": 1
```

Response headers

```
connection: keep-alive
content-type: application/json
date: Tue, 08 Aug 2023 15:51:43 GMT
keep-alive: timeout=60
```

| | |  |
|---|---|---|
| **Despliega el código fuente en GitHub y en Docker Hub** | Despliega el código fuente en GitHub y en Docker Hub | |